New algorithms for parameter-swing reactors

van Noorden, T.L.

2002

NEW ALGORITHMS FOR PARAMETER-SWING REACTORS

VRIJE UNIVERSITEIT

New algorithms for parameter-swing reactors

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. T. Sminia,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op dinsdag 18 juni 2002 om 15.45 uur
in het auditorium van de universiteit,
De Boelelaan 1105

door

Tycho Leonard van Noorden

geboren te Eindhoven

promotoren: prof.dr. S.M. Verduyn Lunel
prof.dr.ir. A. Bliek

Leescommissie:
prof.dr. C.C. Pantelides (Imperial College, London, UK)
prof.dr. D. Roose (Katholieke Universiteit Leuven, België)
dr. W.H. Hundsdorfer (CWI, Amsterdam)
prof.dr. R. Krishna (Universiteit van Amsterdam)
prf.dr. M. van Veldhuizen (Vrije Universiteit, Amsterdam)

Dit proefschrift kwam tot stand onder auspiciën van:

THOMAS STIELTJES INSTITUTE
FOR MATHEMATICS

Het onderzoek waarop dit proefschrift is gebaseerd, is uitgevoerd in de afdeling Analyse van de Faculteit der Exacte Wetenschappen, divisie Wiskunde en Informatica aan de Vrije Universiteit Amsterdam, en in het Instituut voor Technische Scheikunde van de Faculteit der Natuurwetenschappen, Wiskunde en Informatica, afdeling Scheikunde aan de Universiteit van Amsterdam.
Een groot gedeelte van het rekenwerk is uitgevoerd op de computers van SARA Reken- en Netwerkdiensten.

Printed by PrintPartners Ipskamp

Contents

Chapter 1

Introduction

After start-up, every continuous chemical process will go through a transient phase. Subsequent to this transient phase, the process will approach a limiting state in which it will remain during the operating time. The performance of the process in the limiting state is the basis for both the design and optimization of the chemical process. The efficient determination of limiting states of continuous chemical processes is therefore of great importance for the study of chemical processes.

Traditionally, only chemical processes with time invariant operating conditions are designed. We refer to these processes as *steady state operated* processes. Beside these steady state operated processes, there exist more and more chemical processes that are operated under operating conditions that vary cyclically in time. These processes are called *cyclic* processes.

For a steady state operated process, the limiting state is often time invariant. Such a state is referred to as a *steady state*. When modeling a steady state operated process, a time invariant state can often be expressed as the solution of a set of algebraic equations or a system of time independent ordinary differential equations. For the theoretical analysis of limiting states of steady state operated processes, there exists a great number of efficient mathematical and numerical tools.

Because of the periodic forcing, a cyclic process can never reach a time invariant state. The limiting state of a cyclic process is therefore inherently dynamic in nature. The typical limiting state of a cyclic process is a so-called *cycle invariant* or *periodic* state. In terms of model equations, a periodic state can often be expressed as the solution of a two point boundary value problem for a (large) system of differential equations. Because of its time dependent characteristics, the theoretical study and numerical determination of a periodic state is a more comprehensive task than the determination of time invariant, steady states.

In this thesis we consider chemical processes with cyclically varying operation conditions. The efficient determination of their periodic states is the aim of our work. In this first chapter we will give an overview of the current state of affairs and sketch the setting for the research in this thesis. We start to introduce the reader to the basic chemical engineering needed in this work. We will discuss a number of examples of periodic

processes and then consider a simple example of a pressure swing adsorber in more detail. We will use this example to illustrate some of the features and problems that one encounters when analyzing periodic processes. Next we continue with an introduction to the numerical and mathematical setting. We will pay attention to existing numerical tools and methods available for the study of cyclic processes. We show that the efficient numerical determination of periodic states is not a straightforward task. This sets the stage for the research in this thesis, which is devoted to the study and development of efficient methods for the computation of periodic states of cyclic processes. In Chapter 2, we will give an overview of the main results.

1.1 Cyclic Processes

Cyclic processes are well studied in chemical engineering literature. In this section we give a short overview of the most common cyclic processes.

- *Pressure swing adsorber*

  In pressure swing adsorption (PSA) processes, gas mixtures are separated by selective adsorption over a bed of sorbent materials. The cyclic nature of these processes arises from the high pressure adsorption phase and the subsequent low-pressure regeneration phase. According to Yang [64], the PSA process was developed by De Montgareuil and Domine in 1957 and independently by Skarstrom in 1958, The Skarstrom PSA cycle was immediately accepted for commercial use in air drying (see Section 1.2 for further details).

- *Thermal swing adsorber*

  Thermal swing adsorption (TSA) processes are similar to pressure swing adsorption processes and also separate gas mixture, but here the cyclic nature of these processes arises from the low temperature adsorption phase and the subsequent high temperature regeneration phase. Studies of thermal swing adsorbers can be found in work by Davis and Levan [9] and Schork and Fair [48]. There also exist processes that are a combination of PSA and TSA.

- *Reverse flow reactor*

  A reverse flow reactor (RFR) is a packed bed reactor in which the flow direction is periodically reversed in order to trap a hot zone within the reactor. In this way even systems with a small adiabatic temperature rise can be operated without preheating the feed stream. Pioneering work for the reverse flow reactor has been done by Boreskov and Matros [2]. The reverse flow reactor has been mathematically modeled by Eigenberger and Nieken [12].

- *Pressure swing reactor*

  In a pressure swing reactor (PSR), reaction and adsorption occur in the same bed. The adsorption is typically used to purify one of the reaction products. The cyclic nature of a pressure swing reactor arises from the same high pressure adsorption and low pressure regeneration phases as in the pressure swing adsorber. The pressure swing reactor is a relatively new process and has been studied by e.g. Vaporciyan and Kadlec [63], Kirkby and Morgan [25] and Kodde and Bliek [26].

- *Simulated moving bed adsorber/reactor*

  In a simulated moving bed process, the position of the feed and product streams are periodically moving along the sorbent bed. In this way a flow of the solid sorbent, which flows counter-currently to the gas, is mimicked. The aim of a moving bed process is to make more efficient use of the sorbent material. Simulated moving bed processes are studied by e.g. Ruthven and Ching in [44]

1.2 An Example: Pressure Swing Adsorption

1.2.1 Operation of the Pressure Swing Adsorption Process

In this section we will discuss a pressure swing adsorption (PSA) process as an example of a cyclic process . We will first explain how the PSA process is operated. Thereafter we will derive the equations used to model this system.

The PSA system we consider here is the process of pressure swing adsorption of H_2O from air onto alumina. It is designed to separate the H_2O from the air, so that dry air is obtained. This system has been studied by Kvamsdal [30], Smiths and Westerberg [50], Raghavan et al. [40]. The cyclic process is operated in four consecutive steps (see Fig. 1.1 and 1.2). The various events during each of these steps is described below.

- The first step is the *adsorption step*. In this step the carrier gas (air) with a trace of the adsorbate (water), which is to be removed from the carrier gas, is led into the reactor in which the adsorbent (alumina) adsorbs the adsorbate. At the product end of the reactor the gas stream contains (close to) no adsorbate. During this stage of the process the pressure is maintained at a high level.

- Before the adsorbent in the reactor is completely saturated with adsorbate so that it does not adsorb any more adsorbate, the product end of the reactor is closed and the pressure is released at the feed end of the reactor. This is the second step, the *blowdown step*.

- When the pressure has dropped to a sufficient low level, it is maintained at this level and 'clean' carrier gas is led into the reactor at the product end so that the adsorbent in the reactor is purged (i.e. 'cleaned'). This is the third step, the *purge step*.

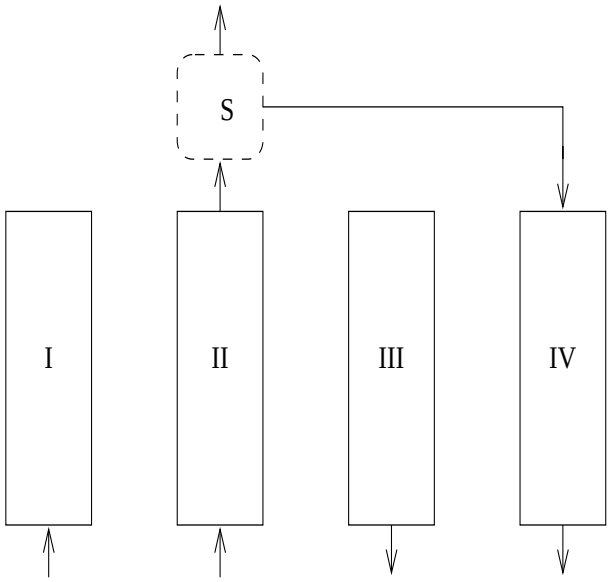

Figure 1.1: The four steps in the operation of the air-drying reactor.

- When the adsorbent has lost enough of its loading, the product end of the reactor is again closed and the pressure is raised to the old high level. This is the fourth step, the *pressurization step*. When the pressure has reached its high level the process switches again to the first step.

A sequence of the above described steps is called a *cycle*. If the reactor is operated for a large number of cycles, the amount of adsorbate that is adsorbed in the adsorption step and the amount of adsorbate that is purged in the purge step will attain an equilibrium and the state of the reactor will converge to a periodic state. This periodic state is sometimes also called a *cyclic steady state* (CSS). This is the state in which the reactor will be operated for most of the time. It is therefore of great important to be able to determine and analyze the possible periodic states of a process.

1.2.2 The Model

In order to be able to find the possible periodic states of a cyclic system, we need a model that describes the behaviour of such a system mathematically. In this section we derive the model equations for the H_2O-air-alumina pressure swing adsorption process. We make the following assumptions:

I The system is isothermal with negligible axial pressure drop. This implies that we do not need temperature as a variable and that the pressure depends on time only. We denote the pressure by $P : [0, \infty) \to \mathbb{R}$.

II Radial dispersion is negligible. This means that the concentration of the gas phase components and the concentrations in the solid phase only depend on time t and the axial direction z. We denote the concentration of the adsorbate in the gas phase by

$C_A : [0,\infty) \times [0,L] \to \mathbb{R}$ and the concentration of the adsorbate in the solid phase by $q_A : [0,\infty) \times [0,L] \to \mathbb{R}$, where L is the bed length. We denote the concentration of the carrier in the gas phase by $C_B : [0,\infty) \times [0,L] \to \mathbb{R}$ and the concentration of the carrier in the solid phase by $q_B : [0,\infty) \times [0,L] \to \mathbb{R}$. We also assume axial dispersion is negligible. This means that we have no diffusion terms in the model equations.

III The adsorbate is a trace component in the carrier gas, which is inert to the adsorbent. This means that the fraction of adsorbate in the carrier is very small so that the change of mass and volume of the gas stream by the adsorption of the adsorbate is negligible. The carrier itself is inert to the adsorbent, i.e., it does not adsorb onto the solid phase and therefore $q_B = 0$.

IV The equilibrium concentration of the adsorbate in the solid phase depends linearly on the concentration of the adsorbate in the gas phase. This assumption can be written as

$$q_A^*(t,z) = KC_A(t,z),$$

where $q_A^* : [0,\infty)\times[0,L] \to \mathbb{R}$ denotes the equilibrium concentration of the adsorbate in the solid phase and where the parameter K is called the adsorption equilibrium constant. We also assume that the mass transfer rate is represented by a linear driving force expression. In formula form we can write this as

$$\frac{\partial q_A}{\partial t}(t,z) - k(P(t))(q_A^*(t,z) - q_A(t,z)) = 0, \quad 0 \le z \le L, t \ge 0. \tag{1.1}$$

This equation is an approximation of the diffusion process of the adsorbate in spherical sorbent particles. The linear driving force rate constant $k(P(t))$ depends on the pressure P. This dependence is given by

$$k(P(t)) = k_{ads}\frac{P_H}{P(t)},$$

where P_H is the high pressure level during the adsorption step. The parameter k_{ads} denotes the linear driving force rate constant during the adsorption step. The explanation for the dependence of the linear driving force rate constant on the pressure is that the diffusion coefficient for binary mixtures of gases varies inversely with the pressure [42].

V The pressure is assumed to be constant during the adsorption and desorption steps.

VI We consider the gas phase to be ideal. This means that we assume that

$$C = C_A + C_B = \frac{P}{RT}, \tag{1.2}$$

where C denotes the total gas phase concentration (moles per volume), R the gas constant (J/mol/K) and T the temperature (K).

VII The flow pattern of the gas phase is assumed to be a plug flow pattern. This means that the velocity of the gas stream does not depend on the radial coordinate. We denote the velocity of the gas stream by $u : [0, \infty) \times [0, L] \to \mathbb{R}$

Equation (1.1) describes the rate of change in concentration in the solid phase. We also need an expression for the rate of change in concentration in the gas phase. In order to derive this expression, we consider a slice of thickness Δz (m) of the reactor between the points z and $z + \Delta z$. The number of moles M (mol) of gas in this slice is given by the law for a perfect gas

$$M = \epsilon A \Delta z \frac{P}{RT},$$

where A (m^2) is the area of the cross section of the reactor, ϵ (-) the void fraction of the packed bed. In our example PSA process the gas phase is a mixture of two components. We want to derive an equation for the concentrations C_A and C_B of these two components. We denote the number of moles of component i ($i = A, B$) in the given slice of the reactor by M_i. This number of moles can only change by concentration fluxes across the borders of the slice and adsorption of the component onto the packed bed:

$$\frac{\partial M_i}{\partial t} = AF_i(z) - AF_i(z + \Delta z) - \frac{\partial Q_i}{\partial t}, \tag{1.3}$$

where Q_i (mol) is the number of moles adsorbed in the given slice and $F(z)$ (mol/m^2/s) the concentration flux at point z. This flux is now given by

$$F_i(z) = \epsilon u(z) C_i(z) - \epsilon D_{ax} \frac{\partial C_i(z)}{\partial z},$$

where D_{ax} (m^2/s) is the axial dispersion coefficient. Note that the dependence on t is suppressed in the notation. After substituting this into (1.3) we obtain

$$\begin{aligned} \frac{\partial M_i}{\partial t} &= \epsilon A u(z) C_i(z) - \epsilon A D_{ax} \frac{\partial C_i(z)}{\partial z} \\ &\quad - \epsilon A u(z + \Delta z) C_i(z + \Delta z) - \epsilon A D_{ax} \frac{\partial C_i(z + \Delta z)}{\partial z} \\ &\quad - \frac{\partial Q_i}{\partial t}. \end{aligned}$$

If we now divide this equation by $A\Delta z$ and let Δz go to zero, the right hand side of the equation becomes the derivative with respect to time of the gas phase concentration (mol/m^3) of component i at point z and the last term of the left hand side becomes the derivative with respect to time of the concentration (mol/m^3) of the adsorbed amount of component i at point x. This last concentration can also be written as the product of the bed density (kg/m^3) and the adsorbed amount $q_i(z)$ of component i per unit mass of the bed packing at point z. Now the equation becomes

$$\epsilon \frac{\partial C_i(z)}{\partial t} = \epsilon \frac{\partial}{\partial z} \left(-u(z) C_i(z) + D_{ax} \frac{\partial C_i(z)}{\partial z} \right) - \rho_b \frac{\partial q_i(z)}{\partial t},$$

where ρ_b (kg/m^3) is the density of the packed bed and $q_i(z)$ the number of adsorbed moles of component i per unit mass of the bed packing at point z.

If we now use the assumption II that we have no diffusion terms, we obtain the following mass balance for the concentration of the adsorbate in the gas phase

$$\frac{\partial C_A}{\partial t}(t,z) + \frac{\partial}{\partial z}(u(t,z)C_A(t,z)) + \frac{1-\epsilon}{\epsilon}\frac{\partial q_A}{\partial t}(t,z) = 0, \quad 0 \le z \le L, t \ge 0. \tag{1.4}$$

This equation relates the rate of change in the gas phase concentration to the mass transport of the adsorbate by convection of the gas stream and by adsorption. The boundary conditions for equation (1.4) will vary periodically and will be discussed in the sequel.

For the gas phase concentration C_B of the carrier gas we have a similar mass balance as (1.4), with the difference that $q_B = 0$. Since the pressure is the parameter that is controlled periodically, it is now convenient to formulate an overall mass balance. This balance is obtained by adding the two mass balances for C_A and C_B and then using relation (1.2). We can write the overall mass balance as

$$\frac{1}{P(t)}\frac{\partial P}{\partial t}(t) + \frac{\partial u}{\partial z}(t,z) + \frac{RT}{P(t)}\frac{1-\epsilon}{\epsilon}\frac{\partial q_A}{\partial t}(t,z) = 0, \quad 0 \le z \le L, t \ge 0.$$

Since we assumed that only a trace amount is separated, the last term of this equations is negligible. The equation then becomes

$$\frac{1}{P(t)}\frac{\partial P}{\partial t}(t) + \frac{\partial u}{\partial z}(t,z) = 0, \quad 0 \le z \le L, t \ge 0. \tag{1.5}$$

The complete mathematical description of the example PSA process is now given by the equations (1.1), (1.4), and (1.5):

$$\begin{aligned} \frac{\partial q_A}{\partial t}(t,z) - k(P(t))(q_A^*(t,z) - q_A(t,z)) &= 0, \\ \frac{\partial C_A}{\partial t}(t,z) + \frac{\partial}{\partial z}(u(t,z)C_A(t,z)) + \frac{1-\epsilon}{\epsilon}\frac{\partial q_A}{\partial t}(t,z) &= 0, \qquad 0 \le z \le L, t \ge 0. \\ \frac{1}{P(t)}\frac{\partial P}{\partial t}(t) + \frac{\partial u}{\partial z}(t,z) &= 0 \end{aligned} \tag{1.6}$$

We will continue with a discussion of the boundary conditions. As mentioned before the reactor is operated in four different stages (see Fig. 1.1). Here we will shortly summarize these four stages.

I The pressurization step in which the pressure increases. The product end of the reactor is closed (duration: t_I, see Figure 1.2).

II The adsorption step in which the pressure is kept constant at a high level. The gas-phase flows through the reactor at a constant speed. The adsorbate is adsorbed onto the solid phase (duration: t_{II}).

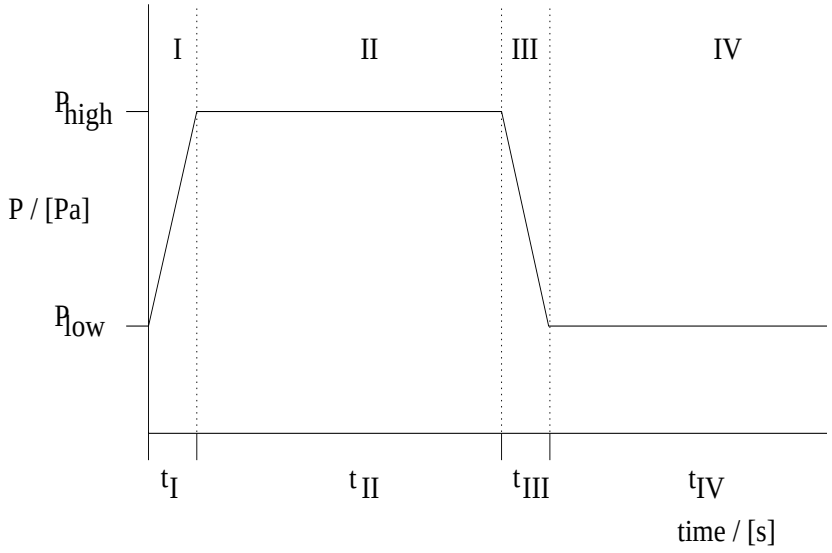

Figure 1.2: The pressure changes periodically with the four different operation steps.

III The blowdown step in which the pressure decreases. The product end of the reactor is closed (duration: t_{III}).

IV The purge step in which the pressure is kept constant at a low level. A purge stream is led into the reactor at the product end. The concentration of the adsorbate in this purge stream can be a constant chosen level or can be related to the concentration in the product stream of another reactor (of the same type) which is in the adsorption step (duration: t_{IV}).

Thus the operating parameters that are periodically changed, are the pressure P (see Fig. 1.2), the velocity of the gas phase u and the adsorbate concentration in the feed and purge streams. To be able to formulate the boundary conditions in a convenient way we define $\bar{t} = t \bmod t_{cyc}$, with $t_{cyc} = t_I + t_{II} + t_{III} + t_{IV}$ the total cycle time and we also define $t_{pres} = t_I$, $t_{ads} = t_I + t_{II}$, and $t_{blo} = t_I + t_{II} + t_{III}$. This yields the following set of boundary conditions

$$\begin{array}{rcll} C_A(t,0) & = & \frac{y_A P(t)}{RT} & 0 \le \bar{t} < t_{ads}, \\ C_A(t,L) & = & 0 & t_{blo} < \bar{t}, \\ u(t,L) & = & 0 & 0 \le \bar{t} < t_{pres}, \\ u(t,0) & = & u_{ads} & t_{pres} \le \bar{t} < t_{ads}, \\ u(t,L) & = & 0 & t_{ads} \le \bar{t} < t_{blo}, \\ u(t,0) & = & u_{pur} & t_{blo} \le \bar{t} < t_{cyc}, \end{array} \tag{1.7}$$

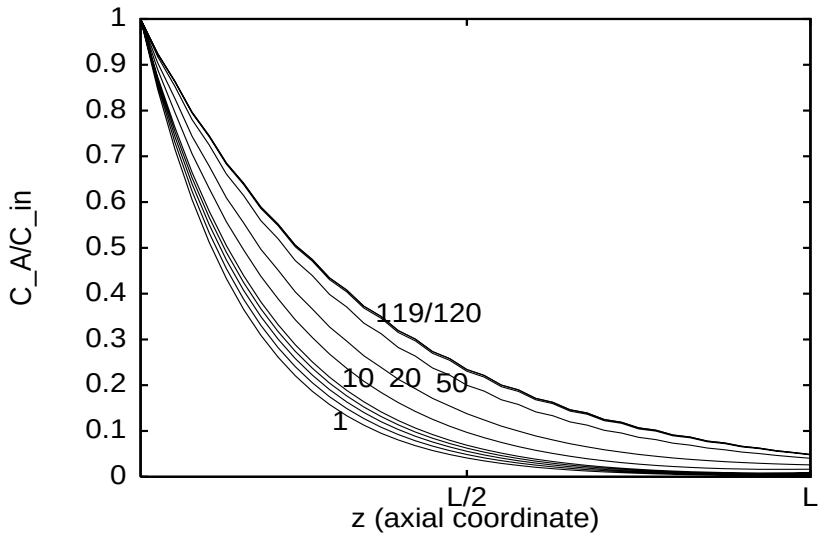

Figure 1.3: The profiles through the bed of C_A after the indicated number of simulated cycles starting with an empty bed. The lowest profile is obtained after simulating one cycle, and the profiles after the five subsequent cycle simulations are also shown. Then the profiles obtained after 10, 20 and 50 cycle simulations are plotted, and the upper two profiles, that are almost indistinguishable, are obtained after 119 and 120 cycle simulations.

where y_A denotes the mole fraction of the adsorbate in the feed gas. The periodic pressure changes are given by

$$P(t) = \begin{cases} P_L + \dfrac{P_H - P_L}{t_{ads}}\bar{t} & 0 \le \bar{t} < t_{pres}, \\ P_H & t_{pres} \le \bar{t} < t_{ads}, \\ P_H - \dfrac{P_H - P_L}{t_{blo} - t_{pres}}(\bar{t} - t_{ads}) & t_{ads} \le \bar{t} < t_{blo}, \\ P_L & t_{blo} \le \bar{t} < t_{cyc}. \end{cases} \tag{1.8}$$

1.2.3 Simulation

In order to simulate the qualitative behaviour of the solutions to the models equations (1.6), we have to numerically integrate these equations. For the numerical integration we first discretize the axial coordinate, using finite differences. In this way we obtain a "large" system of ordinary differential equations. If we supply an initial condition, this system of ODE's can be numerically integrated in time.

In Fig. 1.3, the concentration profiles $C_A(t, z)$ at the end of the adsorption step ($\bar{(t)} = t_{ads}$) are plotted. The numbers indicate the number of cycles simulated before the shown profile is obtained. For the simulation shown in Fig. 1.3, we have discretized the bed using 128 equidistant nodes. Taking the model variables $C_A(t, z)$ and $q_A(t, z)$ into account, we thus obtained a system of 256 ordinary differential equations. The initial condition was an empty bed, i.e., $C_A(0, z) = 0$ and $q_A(0, z) = 0$.

We see that the convergence towards a periodic profile is slow: even after 100 simulated cycles there is a small, but visible change between two subsequent profiles. There exist accounts of cyclic processes that need even many more cycles to converge to a periodic state (see Section 1.5.1). This means that a numerical parameter study or a numerical bifurcation analysis, using the simulation to compute periodic states, will be computationally very expensive. The slow convergence also prohibits the use of more complex models, since the simulation time for one cycle will increase when the model equations become even more complex.

This leads to the need of more efficient methods to compute periodic states of cyclic processes. In the remainder of this chapter, we will discuss in more detail the numeric determination of periodic states and we will give an introduction to the methods proposed in literature.

1.3 The Period Map

The model (1.6) for the example PSA process is formulated using a system of partial differential equations. In order to investigate the behaviour of the solutions of such a model numerically, the model equations have to be discretized. Just as for the simulation in Section 1.2.3, for most of the existing methods, first the spatial variable of the model equations is discretized. There exist many different approaches and variations of how to discretize the model equations. The most commonly used methods in chemical engineering literature are finite difference methods and orthogonal collocation methods. In [40], Raghavan et al. compare both kinds of methods for the simulation of a PSA process. In [53] and [34] a number of different finite difference techniques for PSA processes are discussed.

The discretization of the spatial variable leads to a large system of N (where N depends on the space discretization) time periodic ordinary differential equations. This system can, abstractly, be written as

$$x'(t) = f(t, x) \ , \ \text{ where } \ f(t + T, \cdot) = f(t, \cdot) \ , \tag{1.9}$$

where $x(t)$ is a N dimensional vector, and $f(t, \cdot)$ a function from $\mathbb{R}^N$ to $\mathbb{R}^N$. We do not enter into the details of the spatial discretization. We assume here that the discretization is such that the solutions of the system (1.9) are good approximations to the solutions of the original model system of partial differential equations.

A periodic state of the cyclic process is now equivalent to a T-periodic solution x, i.e., a solution with $x(0) = x(T)$ of this system. The map F that assigns to the initial data at time zero, $x(0)$, the solution after one cycle, at time $t = T$, $x(T)$, is called the *Poincaré* or *period map*: one may write

$$F(x_0) = x(T, x_0), \tag{1.10}$$

where $x(t, x_0)$ denotes the solution of equation (1.9) with initial condition $x(0) = x_0$. The numerical evaluation of the map F requires another discretization step, namely the

numerical integration of the system (1.9) over one period T in time, starting with a given initial condition x_0. It is clear that a T-periodic solution is a solution of the fixed point equation $y = F(y)$. Methods to solve this equation are called shooting methods.

1.4 Stability

Suppose now that $\bar{x}(t)$ is a periodic solution of (1.9). If every solution of (1.9) with initial value to the periodic solution $\bar{x}(t)$, converges eventually to $\bar{x}(t)$, then the solution $\bar{x}(t)$ is called stable. More formally, the solution $\bar{x}(t)$ is called *stable*, if for any $\varepsilon > 0$, there exists a $\delta > 0$ such that if $|x_1 - \bar{x}(0)| < \delta$ then $x(t, x_1)$ exists for all $t \geq 0$ and $|x(t, x_1) - \bar{x}(t)| < \varepsilon$ for all $t \geq 0$. If, in addition $|x(t, x_1) - \bar{x}(t)| \to 0$ as $t \to \infty$, then the periodic solution $\bar{x}(t)$ is called *asymptotically stable*. A periodic state that is not stable is said to be *unstable*.

The stability of a periodic solution is an important property of this solutions, because in practice a cyclic process can never be operated in a state corresponding to an unstable solution of the equation (1.9).

There is an important result that enables us to determine the stability of a periodic solution of (1.9). In order to state this result, we define $M(t)$ to be the matrix $M(t) := \frac{\partial f}{\partial x}(t, \bar{x}(t))$. This matrix is periodic in t with period T. Now let the matrix $X(t)$ be a solution of the linear system

$$X'(t) = M(t)X(t), \tag{1.11}$$

with initial condition $X(0) = I$. This equation is called the variational equation. The eigenvalues of the matrix $X(T)$, are called the *characteristic* or *Floquet multipliers* of $M(t)$. The matrix $X(T)$ itself is called the *monodromy matrix* (see e.g. [16]). Denote the characteristic multipliers of $M(t) := \frac{\partial f}{\partial x}(t, \bar{x}(t))$ by $\lambda_1, ..., \lambda_N$. If now $|\lambda_i| < 1$ for $i = 1, ..., N$, then $\bar{x}(t)$ is an asymptotically stable periodic solution of (1.9).

The monodromy matrix $X(T)$ can be calculated by integrating (1.11), but it can also be obtained via the identity

$$\frac{\partial F}{\partial x}(\bar{x}(0)) = X(T).$$

This equation can be used to compute the monodromy matrix $X(T)$ by finite difference approximations of the Jacobian $J(\bar{x}(0)) := \frac{\partial F}{\partial x}(\bar{x}(0))$ of F at $\bar{x}(0)$, where $\frac{\partial F}{\partial x}(\bar{x}(0))$ is the $N \times N$ matrix whose klth entry is $\frac{\partial F_k}{\partial x_l}$ evaluated at $\bar{x}(0)$.

1.5 Existing Methods

1.5.1 Picard Iteration

The simplest method to compute a fixed point of a map is to iterate the map itself. This method is called Picard iteration. For the map F this can be expressed by

$$x_{i+1} = F(x_i), \quad x_0 \text{ given},$$

where F is defined by (1.10) and where $x_i = x(iT)$ is the solution of the system after i cycles.

If we consider this method from the point of view of equation (1.9), we see that this method can also be regarded as simply integrating (1.9): in order to obtain x_{i+1} we integrate (1.9) over $i+1$ cycles, starting the first cycle with initial condition x_0. Therefore Picard iteration is nothing else than simulating the process in time, just as we have shown in Section 1.2.3. In chemical engineering literature, Picard iteration is sometimes referred to as dynamic simulation [38] or successive substitution [6, 50]. An advantage of the Picard iteration of F is that the physical behaviour of the chemical processes is closely emulated. However, as we have seen in Section 1.2.3, the process may demonstrate a very slow convergence to the periodic state. In this case a large number of "redundant" cycles of the process has to be simulated before a periodic state is reached. This is often the case in the presence of large capacity terms, such as the adsorption capacity or the heat capacity in the case of fixed bed reactors/separators. The other numerical methods presented in this section are developed in an attempt to minimize the number of cycles simulated before convergence.

It is clear that the Picard iteration can only converge to stable periodic solutions. This implies, for example, that the method cannot be used in a branch following procedure to compute unstable periodic states beyond a bifurcation point.

In Fig. 1.4 the convergence to a periodic state for the Picard iteration is depicted for the example PSA process modeled by eq. (1.6), when started with an empty bed as initial condition. On the y-axis the deviation from the periodic state is given and on the x-axis the number of Picard iterations is given. Note that more than 250 iterations are needed in order to reach an accuracy of 10^{-6}.

The number of Picard iterations needed for convergence to a periodic state depends strongly on the cyclic process under investigation. Kikkinides et al. report convergence as fast as four to six cycles [24] or as slow as 10000 cycles [23] for pressure swing adsorption processes. Furthermore, Cheng et al. [6] observe convergence of a pressure swing reactor to a periodic state in 5000 cycles and Rege et al. [41] report convergence in 200–500 cycles for a PSA process.

1.5.2 Newton's Method

If we define the map $G = F - I$, where F is given by (1.10), then fixed points of F become zeros of G. Newton's method [10] generates approximations of a zero of G using the iteration scheme

$$x_{i+1} = x_i - (J(x_i) - I)^{-1} G(x_i) \ ,$$

where $J(x_i) := \frac{\partial F}{\partial x}(x_i)$ again denotes the Jacobian of F at x_i, where $\frac{\partial F}{\partial x}(x_i)$ is the $N \times N$ matrix whose elements are $\frac{\partial F_k}{\partial x_l}$ evaluated at x_i. Thus, $(J(x_i) - I)^{-1}$ is the inverse of the Jacobian of G at x_i. Note that each new iteration requires the computation of J at the previous iteration.

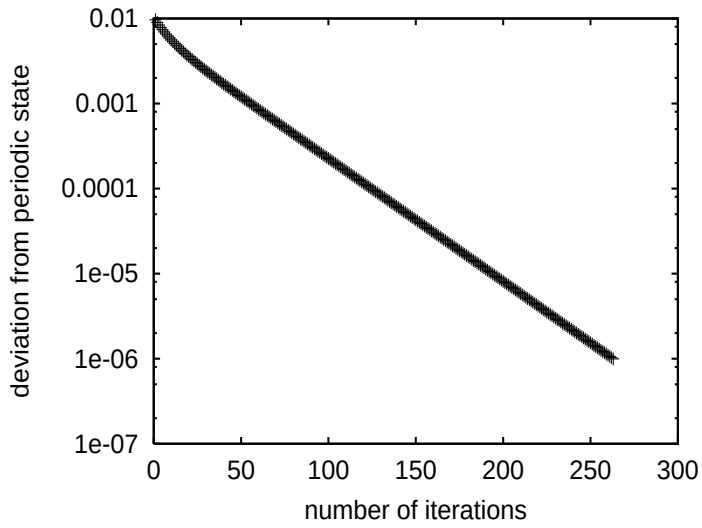

Figure 1.4: The deviation from the periodic state versus the number of evaluations of the period map F for the Picard iteration for the H_2O/air PSA system.

The evaluation of J requires the computation of the dependencies of F on each of the N variables introduced by the discretization of the space dimension. Using first order finite differences to compute these dependencies results in simulating $N + 1$ cycles with different initial conditions. Therefore if the dynamic simulation converges already in less than N cycles, Newton's method will not accelerate the convergence.

For the example PSA system (1.6), where we discretized the axial variable and obtained a system of 256 ordinary differential equations, this means that Newton's methods needs at least 257 evaluations of F. This does not result in a significant shorter computation time, when compared to the Picard iteration, which needed approximately 260 cycles, see Fig. 1.4. Note that the number of evaluations of F for Newton's method depends strongly on the axial discretization, whereas for the Picard method the convergence will be essentially independent of the chosen discretization. For the Picard iteration the number of evaluations of F needed for convergence to a periodic state depend mainly on the properties of the underlying chemical process.

Newton's method is able to compute stable as well as unstable periodic states. The stability of a computed period solutions can be determined by computing the eigenvalues of the Jacobian J at the periodic state. This requires the computation of the Jacobian at the final approximation, but in practice a good idea of the stability can be obtained by looking at the eigenvalues of the Jacobian at the previous approximation, which is readily available.

For Newton's method not only the computation of the Jacobian J is a problem. Another problem is that, when N becomes large, then the storage of the Jacobian (or its inverse), which is an $N \times N$ matrix, might become impossible. For the example PSA system, the Jacobian is not too large to be stored. For more complex model equations, however, for example when the radial dimension of the adsorber would have been incor-

porated, then the Jacobian might easily become too large.

In [7], Croft and Levan propose Newton's method to compute periodic states. They refer to their method as *direct determination*. The Jacobian of the period map of the process needed for the Newton iterations is calculated using the variational equations. This method only accelerates the convergence for systems that converge slowly to a periodic state. The method also allows for the calculation of periodic states satisfying some design constraints. Croft and Levan use the same method in [8] to conduct a bifurcation analysis for a number of example systems. In [11], Ding and Levan use Newton's method in combination with an interpolation technique for the efficient computation of approximations to the Jacobian.

1.5.3 Broyden's Method

Broyden's method is an attempt to avoid computing the Jacobian J in each iteration of Newton's method. Broyden's method [3] starts with an approximation of J which is updated each iteration. To be more precise, Broyden's method produces approximations to a zero of G using the following iteration scheme

$$x_{i+1} = x_i + H_i G(x_i) \ , \tag{1.12}$$

with H_i iterative approximations to $-(J(x_i) - I)^{-1}$ defined by

$$H_{i+1} = H_i - \frac{(p_i + H_i g_i) p_i^T H_i}{p_i^T H_i g_i} \ , \tag{1.13}$$

where $g_i = G(x_{i+1}) - G(x_i)$ and $p_i = x_{i+1} - x_i$. The only information that is used in updating H_i is the function value of G at the new iteration. The method thus uses only one cycle simulation in each iteration. This is a large advantage in comparison to Newton's method, which uses $N + 1$ cycle simulations each iteration. If we assume that the evaluation of F is computationally more expensive than the computational work in eq. (1.12) and (1.13), then the amount of computational work of one iteration of Broyden's method is comparable to the amount of work of an iteration of the dynamic simulation. As initial approximation, the choice $H_0 = I$ is in many cases a robust choice [11]. With this initial approximation the first iteration of Broyden's method is identical to a Picard iteration.

In Fig. 1.5 we show the convergence of Broyden's method for the PSA process discussed in Section 1.2. We see that Broyden's method needs less than 35 evaluations of F, whereas both the Picard iteration and Newton's method need more than 250 evaluations of F.

Like Newton's method, Broyden's method is able to compute stable and unstable periodic states. For Broyden's method, however, it is not true that the approximations H_i converge to the inverse of the Jacobian of G [10]. The final H_i, therefore, cannot be used for the determination of the stability of the computed periodic state.

Broyden's method as described in eq. (1.13), needs to store, just as Newton's method, an $N \times N$ matrix, namely H_i. When N is large, this might be impossible. As an alternative

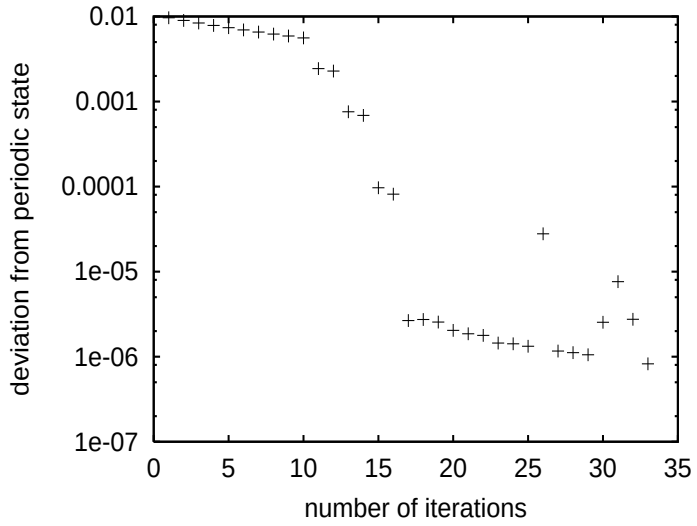

Figure 1.5: The deviation from the periodic state versus the number of evaluations of the period map F for Broyden's method for the H_2O/air PSA system.

to eq. (1.13), we could start with $H_0 = I$ and then store each subsequent rank one update using two vectors. With this procedure, the needed storage depends on the number of iterations that Broyden's method needs to converge. If this number becomes too large, then one has to decide which updates to keep. For a possible approach to using Broyden's method with limited storage, see [39].

Smith IV and Westerberg [50] use Broyden's method to compute periodic states of a linear PSA process. They also propose a method for choosing the initial H_0. With this method acceleration factors ranging from roughly 3 to 50, when compared to Picard iteration, could be reached.

In [52], Stepanek et al. use Broyden's method and several other quasi-Newton methods for the determination of periodic states of a dessicant rotor. Khinast et al. [22, 21] use Broyden's method for the computation of branches of periodic states of reverse flow reactors. They also use the method to continue codimension-1 bifurcation points.

1.5.4 Double Discretization Method

The double discretization method is a method that is, unlike the previously discussed methods, not a shooting method. The strategy of the method is to discretize the axial and time axis simultaneously using finite element or finite difference methods. The periodic boundary conditions are imposed directly. In this way a large system of algebraic equations is obtained that is solved using a standard method. The method is also referred to as the complete discretization method [38], the relaxation method [22] or the global discretization method [56].

Gupta and Bhatia [15] use the method to compute periodic states of a reverse flow reactor and report significant savings of computer time compared to Picard iteration. They use a finite element method for the spatial discretization and a finite difference

method for the temporal discretization. In [46], Salinger and Eigenberger use the double discretization method with finite element techniques for the determination of periodic states of reverse flow reactors. In [47], the same authors explain how the method can be used for a bifurcation analysis for the considered processes. Nilchan and Pantelides [38] and Cheng et al. [6] use the double discretization method for the optimization of pressure swing adsorption processes and a pressure swing reactor, respectively.

The main drawback of the relaxation method, as observed by Unger et al. in [56] and Khinast and Luss in [22], is the large computer memory it requires. This is related to the fact that the size of the Jacobian, which needs to be calculated if a Newton type algorithm is used to solve the large system of algebraic equations, grows quadratically with the number of equations. This drawback would disappear if a more storage-efficient method is used to solve the system of algebraic equations, such as a multigrid method.

In this thesis the double discretization method is not considered. The main reason being the disadvantage stated above. Another reason is the fact that it is not easy to make a good comparison between the double discretization method and the above discussed shooting methods. It is difficult to compare the accuracy of the solutions obtained by the different methods, since the time discretization is completely different for the double discretization method and a shooting method.

1.5.5 Other Methods

In this section we give some references to papers in which periodic processes are studied using methods that are not related to the methods discussed above.

For some cyclic processes the model equations allow an analytical study of the periodic state. In [54], Sundaram presents an analytical solution for the periodic state for a simple PSA process. Bhatia uses in [1] a perturbation method in combination with a numerical procedure to study periodic states of a reverse flow reactor.

Lewandowski et al. present an unusual method in [33]: these authors use neural networks for the optimization of a PSA process.

Chapter 2

Outline and Overview of Main Results

We have seen that the efficient computation of periodic states of cyclic processes has been given a reasonable amount of attention in the literature. There exist a number of methods alternative to Picard iteration, and considerable speed up of the computation time when compared to Picard iteration, has been reported. There remain, however, some questions unanswered.

In the literature, especially Broyden's method has been applied with success for the fast computation of periodic states. A natural question is: why is Broyden's method so successful? And is it possible, for a given model of a cyclic process, to decide a priori which method will be most efficient for the computation of the possible periodic states? For some processes, the Picard iteration is not a good choice, but in other cases the Picard iteration may converge in only a few cycles, as we have seen in Section 1.5.1. To answer these questions, the different methods need to be thoroughly compared. In an attempt to give some answers, Kvamsdal and Hertzberg compare in [31] Picard iteration, Broyden's method and Newton's method using two linear PSA example processes. These authors report that Broyden's method is most efficient in most cases, but do not give an explanation for this fact. In [56], Unger at el. also discuss the pro's and con's of various methods. Newton's method using variational equations to calculate the Jacobian is selected as the preferred method to compute periodic states of a PSA system and of a reverse flow reactor.

Another question one could ask is: can we use properties of the underlying chemical processes to construct more efficient methods? Newton's method works for a large class of problems and this class is much larger than the class generated by solving for a periodic solution of a system of time periodic partial differential equations that model a chemical process. In other words, we do not need a method that is as broadly applicable as Newton's method, if we only want to compute periodic states of cyclic chemical processes.

When a cyclic process is studied numerically, we usually require not just the computation of one periodic state, but of a large number of periodic states. We are interested in computing periodic states for different sets of parameter values, in order to, for example,

find an optimal set of parameter values. Therefore it is important that methods for computing periodic states are also efficient in combination with a optimization or continuation procedure.

In a number of papers, cyclic processes with multiple periodic states are reported. This means that there exist for a given set of parameters more than one periodic states. Khinast et al. give in [22, 21] an overview of the multiple periodic states in a reverse flow reactor. In [41, 55] multiple periodic states are reported in PSA processes. It is unknown what the exact conditions are for multiple periodic states to appear in a cyclic process.

In this thesis we discuss the above mentioned issues. We do not pretend to offer a complete solution, but we gain some insight and make a contribution to the final answer to the posed questions.

In Chapter 3 we give numerical evidence that the considered cyclic chemical processes exhibit low dimensional dynamics. By this we mean that the number of large eigenvalues of the Jacobian $J(x)$ of F at various points x is small, and essentially independent of the discretization. Therefore the interesting dynamics, which determines the stability, only occurs in a low dimensional subspace of the whole state space. For a typical plot of the eigenvalues of $J(x)$, see Fig. 2.1. From this plot we also observe that all eigenvalues are real. Another observation (see Fig. 3.8 in Chapter 3) is that especially the smaller eigenvalues of the Jacobian $J(x)$ do not vary much for different parameter values and different states x.

We will introduce a single shooting method in combination with a hybrid Newton-Picard method developed by Lust et al. in [43, 37], which exploits the low dimensional dynamics of the chemical processes for the efficient determination of their periodic states. The Newton-Picard method uses the fact that there are only a small number of large eigenvalues, and avoids the explicit calculation of the Jacobian J of F. We only need to compute the action of J on a p dimensional subspace of $\mathbb{R}^N$, where p is much smaller than N. We discuss the method and apply it to the computation of periodic states of a rapid pressure swing adsorber and a cooled reverse flow reactor. As an illustration of the efficiency of the method we compute a branch of periodic solutions of the reverse flow reactor.

In Chapter 4 we compare the performance of the Newton-Picard method with the performance of Picard iteration, Newton's method and Broyden's method. We apply all four the methods to five basic, well documented and studied models of periodic processes. In order to compare the different methods, two different convergence criteria are introduced, one based on the residual of a state and the other based on the deviation of a state.

Two of the presented systems (H_2O/air and CO_2/He PSA separation) exhibit linear behaviour. For these systems we can make predictions about the performance of the different methods.

It is found that the Newton-Picard method is in most cases much more efficient than either dynamic simulation or Newton's method. Broyden's method is in the tested cases always the most efficient in terms of needed cycle simulations. However, it is shown that Broyden can suffer from robustness problems in nonlinear cases.

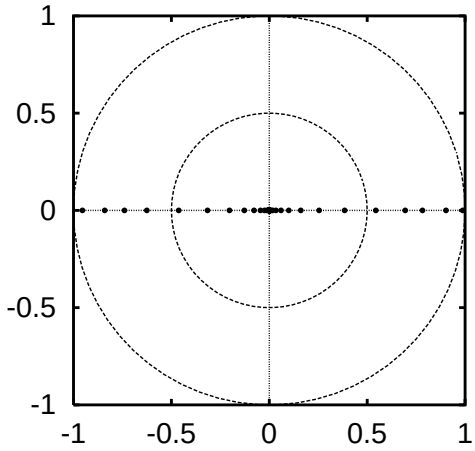

Figure 2.1: A typical picture of the eigenvalues of the Jacobian $J(x^*)$ at the periodic state x^* of the (discretized) infinite dimensional period map F plotted in the complex plane. The underlying cyclic process is a reverse flow reactor. Note that most eigenvalues are very close to zero and only a few eigenvalues are near the unit circle. Also note that all eigenvalues are real.

As mentioned before, the comparison of Broyden vs. Newton-Picard is not a fair one. The Newton-Picard method gives more information about the studied systems than Broyden's method, as the Newton-Picard method also computes approximations to the eigenvalues that determine the stability of the computed periodic states. Thus in situations where the stability of periodic states is needed the Newton-Picard method is clearly a better choice than Broyden's method.

Chapter 5 is devoted to the efficient branch following for periodic states of cyclic processes. We use a pseudo-arclength continuation procedure and we wish to monitor the stability of the periodic states along the computed branch. The Newton-Picard method is very well suited for this job. Broyden's method as such does not determine the stability of periodic states. We are able, however, to adopt Broyden's method, using ideas from the Newton-Picard method, so that it becomes suited to determine the stability of periodic states. We use this new method, which we call the BSI rank $p+1$ method, in combination with the pseudo-arclength procedure and compare its performance with the performance of the Newton-Picard method. Using both methods we compute a branch of periodic states of a reverse flow reactor. This branch is depicted in Fig. 2. It turns out that the BSI rank $p+1$ method is the most efficient method of the two.

In Chapter 6 we consider the optimal control of a rapid pressure adsorber and a rapid pressure swing reactor. We use a first order gradient method for the solution of the optimal control problem. We don not attempt to answer the question which method for the computation of periodic states is most efficient in combination with this gradient method. We only show that the combination of the Newton-Picard method with the first order gradient method is very efficient.

The solutions to the optimal control problems show that for both the rapid pressure swing adsorber and the rapid pressure swing reactor the optimal cyclic operation scheme

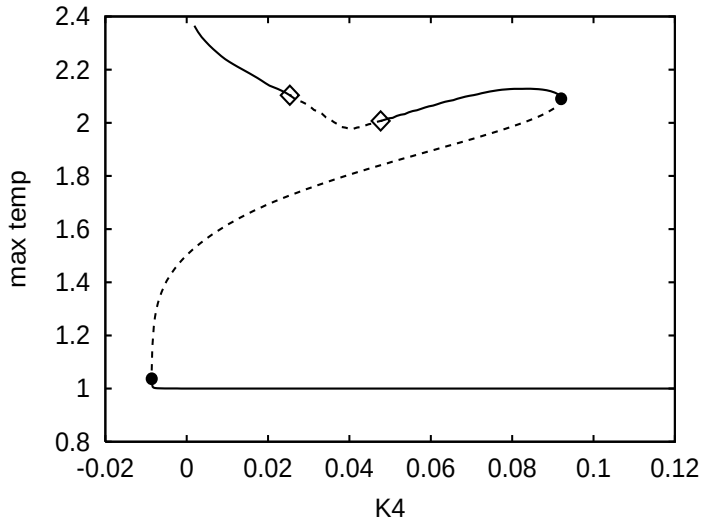

Figure 2.2: Branch of periodic solutions for the cooled reverse flow reactor. Along the horizontal axis the parameter K_4, which is a measure for the cooling capacity, is plotted and along the vertical axis the maximal dimensionless temperature in the bed at flow reversal is plotted. solid line: stable periodic states, dashed line: unstable states. $\diamond$: an eigenvalue crosses the unit circle at -1, $\bullet$: an eigenvalue crosses the unit circle at 1.

consists of four steps instead of the two steps in the original operation scheme. This is illustrated in Fig. 2.3, where it is shown that the original pressurization step is replaced by three different steps, two steps in which the feed end is closed, and an intermediate pressurization step. This optimal operation scheme requires the continuous control of the feed pressure in time. We show that this optimal scheme can be well approximated by a four step peace-wise constant parameter control scheme for the feed pressure (see Fig. 2.4).

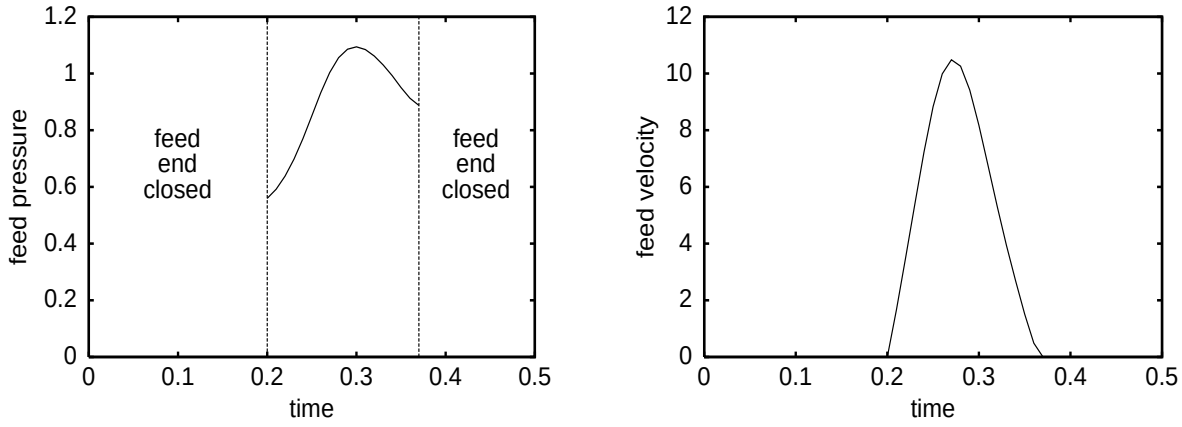

Figure 2.3: The dimensionless feed pressure (left figure) and the dimensionless feed velocity (right figure) versus the dimensionless time for the optimal time-programmed feed velocity for the rapid pressure swing adsober. A purity of 87.0 % is obtained and the performance index equals 36.87. See Chapter 6 for more details.

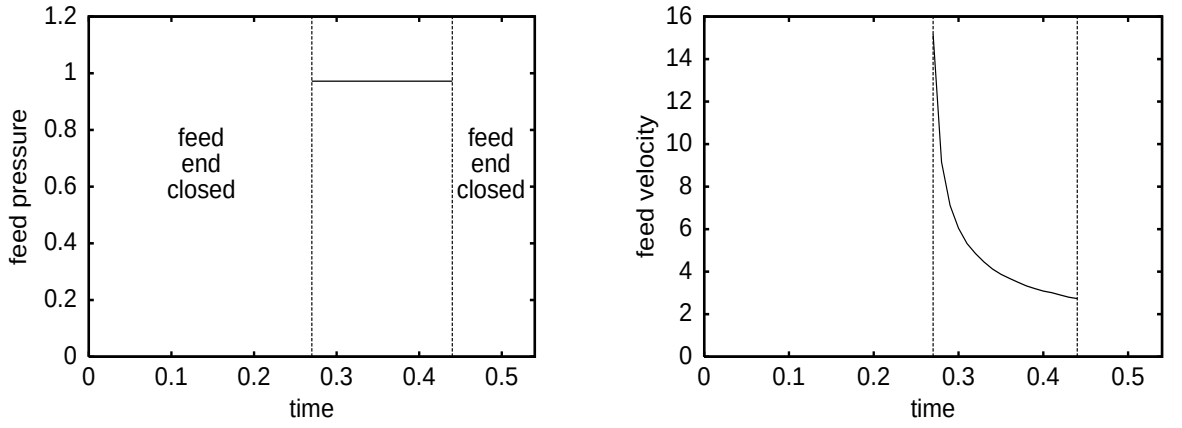

Figure 2.4: The dimensionless feed pressure (left figure) and the dimensionless feed velocity (right figure) versus the dimensionless time for the optimal four step peace-wise constant parameter control scheme for the feed pressure. A purity of 87.0 % is obtained and the performance index equals 41.0. See Chapter 6 for more details.

Chapter 3

A Newton-Picard Method for Periodically Forced Systems

3.1 Introduction

In the previous chapters, we saw that the dynamics of a periodic process can be simulated by iterating the period map $F : \mathbb{R}^N \to \mathbb{R}^N$ corresponding to the process. The dimension N depends on the space-discretization of the model equations. A fixed point of the map F corresponds to a periodic state of the process. The methods that we discussed in Section 1.5 for finding fixed points of F, are designed to work for *general* periodically forced systems of partial differential equations. None of the methods uses any characteristic of the underlying chemical processes.

In this chapter we give numerical evidence that the chemical processes under consideration exhibit low dimensional dynamics. By this we mean that the number of large eigenvalues of the Jacobian $J(x)$ of F at various points x is small, and essentially independent of the discretization. Therefore the interesting dynamics, which determines the stability, only occurs in a low dimensional subspace of the whole state space. For typical plots of the eigenvalues of $J(x)$, see Figures 3.3 and 3.6 in Section 3.4. From these plots we also observe that almost all eigenvalues are real. Another observation (see e.g. Fig. 3.8) is that especially the smaller eigenvalues of the Jacobian $J(x)$ do not vary much for different parameter values and different x.

We will discuss a single shooting method in combination with a hybrid Newton-Picard method that uses the low dimensional dynamics of the chemical processes for the efficient determination of their periodic states. The Newton-Picard method uses the fact that there are only a small number of large eigenvalues, which means that we can avoid the explicit calculation of the Jacobian J of F, when we want to use a Newton-like method. We only need to compute the action of J on a p dimensional subspace of $\mathbb{R}^N$, where p is much smaller than N.

We will apply the presented variant of the Newton-Picard method for periodically forced systems to compute periodic states of a rapid pressure swing adsorption system

and a cooled reverse flow reactor. As an illustration of the efficiency of the method we compute a branch of periodic solutions of the reverse flow reactor.

The organization of this chapter is as follows. In Section 3.2, we present the Newton-Picard method. In Section 3.3, we discuss the models for the rapid pressure swing adsorber and the reverse flow reactor. In Section 3.4, we show that the cyclically operated chemical processes introduced in Section 3.3 exhibit low dimensional dynamics and we compute the periodic states of these processes using the Newton-Picard method.

3.2 The Newton-Picard Method

The Newton-Picard method is a Newton-type method. This means that the method produces approximations to a fixed point of F, using the iteration scheme

$$\begin{cases} x_{i+1} = x_i + \Delta x_i \text{ with} \\ \Delta x_i \approx -(J(x_i) - I)^{-1}(F(x_i) - x_i), \end{cases} \quad , \quad i = 0, 1, 2, ... \tag{3.1}$$

where $J(x_i)$ denotes the Jacobian of F at x_i and where x_0 is a given initial approximation.

We will explain one iteration of the method and so we fix i and set $x = x_i$. Assume that approximations μ_1, μ_2, ..., μ_N to the eigenvalues of $J(x)$ are known, ordered by decreasing modulus: $|\mu_1| \geq |\mu_2| \geq ... \geq |\mu_p| \geq |\mu_{p+1}| \geq ... \geq |\mu_N|$. We also need a parameter ρ of the method that has to be chosen between 0 and 1. This parameter ρ determines a number p that is such that

$$|\mu_1| \geq |\mu_2| \geq ... \geq |\mu_p| > \rho > |\mu_{p+1}| \geq ... \geq |\mu_N|.$$

Define U to be the subspace of the discretized state space $\mathbb{R}^N$ spanned by the eigenvectors and generalized eigenvectors of $J(x)$ corresponding to the eigenvalues μ_k, $k = 1, ..., p$. Let $V_p \in \mathbb{R}^{N \times p}$ denote an orthogonal basis of U, i.e., the columns of V_p are orthogonal and span the subspace U. Finally, let $V_r \in \mathbb{R}^{N \times (N-p)}$ be an orthogonal basis of $U^\perp$, the orthogonal complement of U in $\mathbb{R}^N$.

The idea is to decompose the correction Δx into a component in U and a component in $U^\perp$. For this decomposition we construct the orthogonal projectors $P := V_p V_p^T$ onto U, where V_p^T denotes the transpose of the matrix V_p and $Q := V_r V_r^T = I - P$ onto $U^\perp$. Consequently, the correction can be written as

$$\Delta x = P\Delta x + Q\Delta x = V_p V_p^T \Delta x + V_r V_r^T \Delta x = V_p \Delta\bar{p} + V_r \Delta\bar{r}, \tag{3.2}$$

where the notation $\Delta\bar{p} := V_p^T \Delta x$ and $\Delta\bar{r} := V_r^T \Delta x$ is used.

We would like Δx to be an approximate solution of

$$(J(x) - I)\Delta x = -(F(x) - x). \tag{3.3}$$

So we substitute (3.2) in (3.3) and multiply the left and right hand side of (3.3) by $(V_p \;\; V_r)^T$. This leads to the matrix equation

$$\begin{pmatrix} V_p^T(J(x) - I)V_p & V_p^T J(x) V_r \\ 0 & V_r^T(J(x) - I)V_r \end{pmatrix} \begin{pmatrix} \Delta\bar{p} \\ \Delta\bar{r} \end{pmatrix} = - \begin{pmatrix} V_p^T(F(x) - x) \\ V_r^T(F(x) - x) \end{pmatrix}. \tag{3.4}$$

The lower left block of the matrix in the left hand side of (3.4) equals $V_r^T(J(x) - I)V_p$ and this equals zero, because the subspace spanned by V_p is invariant under $J(x)$ and is orthogonal to the subspace spanned by V_r. Because we would like to avoid the computation of the matrix in the left hand side of (3.4), we can use the fact that the last row of this equation does not contain an expression in $\Delta\bar{p}$. Therefore this row (which contains $N - p$ equations) can be solved for $\Delta\bar{r}$ with the Picard iteration scheme

$$\Delta\bar{r}_{j+1} = V_r^T J(x)V_r\Delta\bar{r}_j + V_r^T(F(x) - x). \tag{3.5}$$

For this iteration scheme we do not need to compute $J(x)$, we only need to be able to compute the matrix-vector products $J(x)V_r\Delta\bar{r}_j$. Since we actually would like to compute $V_r\Delta\bar{r}$, it is more convenient to iterate

$$(V_r\Delta\bar{r})_{j+1} = V_rV_r^T J(x)(V_r\Delta\bar{r})_j + V_rV_r^T(F(x) - x). \tag{3.6}$$

We now can also avoid computing and using the matrix V_r by substituting the identity $V_rV_r^T = I - V_pV_p^T$ into (3.6):

$$(V_r\Delta\bar{r})_{j+1} = (I - V_pV_p^T)(J(x)(V_r\Delta\bar{r})_j + (F(x) - x)). \tag{3.7}$$

Because of the way the two subspaces U and $U^\perp$ are constructed, the scheme (3.7) will converge much faster than the Picard iteration of the period map F.

If the approximation of fixed point of (3.7) is again denoted by $V_r\Delta\bar{r}$, then the equation for $\Delta\bar{p}$ becomes

$$V_p^T J(x)V_r\Delta\bar{r} + V_p^T(J(x) - I)V_p\Delta\bar{p} = -V_p^T(F(x) - x).$$

In order to solve this equation, we only need to know the action of the Jacobian $J(x)$ on the p-dimensional subspace spanned by V_p. This means we only need to compute the p matrix-vector products $J(x)V_p$, instead of computing the whole Jacobian $J(x)$.

Together $V_r\Delta\bar{r}$ and $V_p\Delta\bar{p}$ make up the update in the scheme (3.1), which, with the decomposition made in (3.2), can be written as

$$x_{i+1} = x_i + V_p\Delta\bar{p} + V_r\Delta\bar{r}.$$

The dimension p of U and an initial approximation to a basis V_p of the subspace U is obtained by computing, before the first Newton-Picard iteration, the Jacobian with finite differences and calculating its eigenvalues and eigenvectors. The eigenvectors and generalized eigenvectors corresponding to the p largest eigenvalues span the subspace U. An orthogonal basis can be obtained using a Gram-Schmidt orthogonalization procedure. Notice that since the Jacobian is available for this first iteration, we can take the first iteration to be a Newton iteration. In the subsequent iterations of the Newton-Picard algorithm, the basis of the subspace U is updated with one or more iterations of the subspace method with projection. It is important to note that in this subspace iteration, approximations to the p largest eigenvalues of the Jacobian are obtained. These eigenvalues are essential in the determination of the stability of a periodic state. The version

of the subspace iteration algorithm that we use, is the version also used in [37]. The algorithm can be summarized by the following steps.

Subspace iteration with projection

Step 1: *Set* $\tilde{V}_0 = V_p$ *and set* $j = 0$

Step 2: *Compute* $W_j = \frac{\partial F}{\partial x}(x_1)\tilde{V}_j$.

Step 3: *Compute* $S_j = \tilde{V}_j^T W_j$.

Step 4: *Compute the Schur vectors* $Y_j = [y_1, ..., y_p]$ *of* S_j*, order them according to decreasing modulus of the corresponding eigenvalue.*

Step 5: *Compute* $\tilde{V}_{j+1} = W_j Y_j$.

Step 6: *Orthonormalize* $\tilde{V}_{j+1}$.

Step 7: *If* $\tilde{V}_{j+1}$ *is accurate enough, then stop and return* $\tilde{V}_{j+1}$ *and* W_j*. Else set* $j = j+1$ *and go to* **Step 2**.

In this procedure the matrices $\tilde{V}_j$ will converge to a orthonormal basis of the subspace spanned by the eigenvectors and generalized eigenvectors associated with the p eigenvalues of $J(x_i)$ with the largest modulus.

In contrast to the approach in [37], where the subspace iteration is performed on $p + p_e$ (with $p_e = 2$, 3 or 4) vectors in order to accelerate the convergence of the subspace iteration procedure and to improve the detection of eigenvalues leaving or entering the region $C_\rho := \{z \in \mathbb{C} : |z| < \rho\}$, we only perform the subspace iteration on p vectors, and fix the dimension of the subspace U before the first iteration of the Newton-Picard algorithm. In this way we save some computation time, since the subspace iteration is performed on a smaller dimensional subspace. This approach, however, creates of course the problem that, at a certain point in the iteration procedure, eigenvalues may exist that either:

1) leave C_ρ, but the corresponding eigenvector is not yet in the subspace U or

2) enter C_ρ, but the corresponding eigenvector is still in U.

Case 2) is not a real problem: it will only cause the Picard iteration in eq. (3.5) to converge even faster. In Case 1), the Picard iteration in eq. (3.5) slows down and in the worst case, when an eigenvalue has moved unnoticed out of the unit disk C, it will not converge at all. In both Case 1) and 2) also the possibility exists that a complex conjugated pair of eigenvalues becomes larger in absolute value than the smallest (real) eigenvalue of which the corresponding eigenvector is in U. In this case the subspace iteration procedure will experience convergence problems, since the eigenvector in U belonging to this smallest (real) eigenvalue will try to converge to the two dimensional eigenspace belonging to the conjugated pair of eigenvalues.

The above described problems can be overcome by a procedure that monitors the convergence rate of the Picard iterations in eq. (3.5) and adjusting the dimension of U accordingly. This approach is used in [49] and [19]. In our computations for the systems that will be described in Section 3.3, we did not need to use such a monitoring procedure. For these systems, the eigenvalues with modulus close to ρ change only very little during the iterations of the Newton-Picard method (see also Fig. 3.8), so that the number of eigenvalues inside C_ρ stays constant. These observations motivated our choice to fix the dimension of the subspace U before the first Newton-Picard iteration. We are aware that this choice cannot be justified for general nonlinear periodic systems of differential equations. For systems where the eigenvalues with modulus close to ρ change rapidly, a monitoring procedure as used in [49] and [19] or the approach in [37] is necessary. Our aim is, however, to construct a method dedicated to the computation of periodic states of cyclic processes. The method presented here is based on the observed properties of the studied cyclic processes.

Another difference between the method outlined above and the method in [37] is that we do not use locking and deflation in the subspace iteration and that the matrix-vector products $J(x_i)v_j$, $J(x_i)V_{p_i}$ and $J(x_i)(V_r\Delta\bar{r})$ in **Step 6, 7** and **8** in the algorithm below, are calculated using a finite difference approximation given by

$$J(x)v \approx \frac{||v||}{\epsilon}(F(x + \frac{\epsilon}{||v||}v) - F(x)).$$

The Newton-Picard algorithm as implemented, performs a fixed number of iterations of the Picard scheme (3.5). This number is denoted by l in the following description of the steps of the algorithm. The number δ_0 in **Step 5** is the bound for convergence to a periodic state. In parenthesis, we denote the number of evaluations of F that is needed.

Newton-Picard Algorithm

Step 1: *Choose an initial value x_0 and set $i = 0$.*

Step 2: *Compute the Jacobian $J(x_0)$ of F at x_0 (finite differences: $N+1$ evaluations of F).*

Step 3: *Compute the eigenvalues and eigenvectors of $J(x_0)$. This determines p, the dimension of the slow subspace U and an orthonormal basis V_{p_0} of U.*

Step 4: $x_1 = x_0 - (J(x_0) - I)^{-1}(F(x_0) - x_0)$.

Step 5: *If $|F(x_i) - x_i| < \delta_0$ then stop (1 evaluation of F).*

Step 6: *Set j=1 and iterate s times: (sp evaluations of F).*

1) *compute the ordered real Schur-factorization $Y^T RY$ of $V_{p_i}^T S$, with $S = J(x_i)V_{p_i}$.*

2) *if $j < s$ then: orthonormalize $V = SY$ and put the result in V_{p_i}. Set $j = j + 1$. (If $j = s$, then this step is postponed until* **Step 10**.*)*

Step 7: $v_0 = (I - V_{p_i} V_{p_i}^T)(F(x_i) - x_i)$. *Iterate* $v_{j+1} = (I - V_{p_i} V_{p_i}^T)(J(x_i)v_j + (F(x_i) - x_i))$ l *times. Set* $V_r \Delta\bar{r} := v_l$. *(Computation of* $J(x_i)v_j$*:* l *evaluations of* F*).*

Step 8: *Solve* $(V_{p_i}^T S - I)(\Delta\bar{p}) + V_{p_i}^T J(x_i)(V_r \Delta\bar{r}) = -V_{p_i}^T (F(x_i) - x_i)$ *for* $\Delta\bar{p}$*. (Computation of* $J(x_i)(V_r\Delta\bar{r})$*: 1 evaluation of* F*).*

Step 9: *Set* $x_{i+1} := x_i + V_{p_i}\Delta\bar{p} + V_r\Delta\bar{r}$.

Step 10: *Compute* $V = SY$*, orthonormalize* V *and put the result in* $V_{p_{i+1}}$*. Set* $i = i + 1$*. Go to* **Step 5**.

The above algorithm amounts to $sp + 2 + l$ evaluations of F per iteration. In our implementation we take $l = 2$ and $s = 1$. Thus, for each iteration of the Newton-Picard method, we need $p + 4$ evaluations of F.

We will end this section with a remark about the extreme values, 0 and 1, of the parameter ρ. The parameter ρ determines the dimension of the subspace U. If we set ρ equal to one for determining a stable fixed point of F, then U is empty and the Newton-Picard method reduce to Picard iteration of the map F. On the other hand, if we set ρ equal to 0, then U equals the whole of $\mathbb{R}^N$, and the Newton-Picard method reduces to Newton's method.

3.3 Two Example Systems

In this section we discuss the models of a reverse flow reactor and of a rapid pressure swing adsorption system. The parameters in this section are all dimensionless. The relation between the dimensionless and the physical parameters are given in Sections A.1 and A.2 in Appendix A.

3.3.1 The Rapid Pressure Swing Adsorber

Pressure swing adsorption processes are systems that separate mixtures of gases by periodically varying the pressure over an adsorption bed (See Fig. 3.1). Among the pressure swing adsorption processes, the Rapid Pressure Swing Adsorption process (RPSA) offers one of the simplest configurations. The RPSA cycle comprises two basic steps: pressurization by feed gas and countercurrent depressurisation with internal purging. Both steps have a short time duration, in the order of several seconds. The adsorption bed consists of small adsorbent particles with average size between 200-700 μm. The effectiveness of self purging within the RPSA bed results from a combination of a fast cycle time together with a small particle size leading to steep and periodically varying pressure gradients within the bed. The pressure at the product end is approximately constant over time and this makes the process useful for continuous product release.

The specific process of interest to this case study is air purification using an adsorption bed of zeolite A5 which preferentially adsorbs nitrogen and leaves oxygen in the product

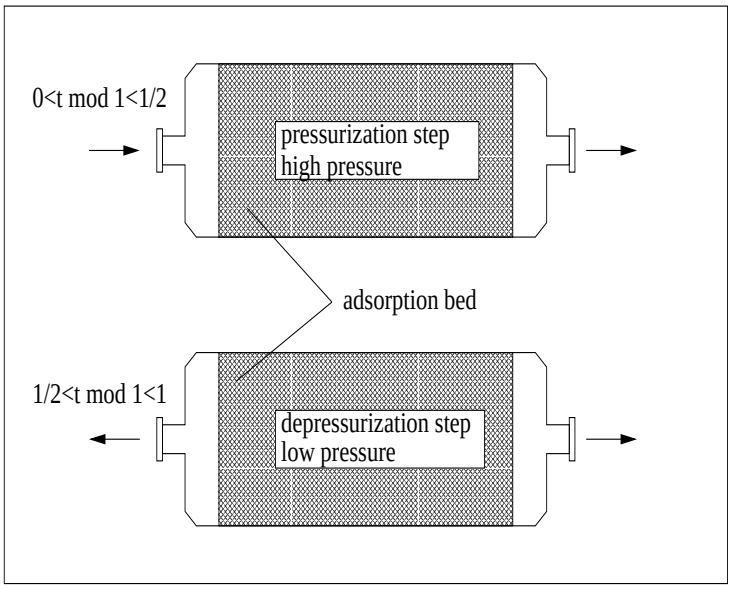

Figure 3.1: A rapid pressure swing adsorber.

stream. The model gas consists of two components, oxygen and nitrogen. The variables in the model are the gas phase concentrations of these two components, denoted by $c^{(i)} : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to c^{(i)}(t,z)$ ($i = 1$: oxygen, $i = 2$: nitrogen), the adsorbed amount of the two components, denoted by $q^{(i)} : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to q^{(i)}(t,z)$ and the gas phase velocity $v : [0,\infty) \times [0,1] \to \mathbb{R},\ (t,z) \to v(t,z)$. The dimensionless parameters that appear in the equations are denoted by C_1, C_2, $C_3^{(i)}$, C_4, C_5, $k^{(i)}$, $D^{(i)}$, $y_f^{(i)}$, p_w and t_f. Their values are listed in Table 3.1. The model equations consist of the gas phase balance equations

$$c_t^{(i)} = C_1 c_{zz}^{(i)} - C_2 (v c^{(i)})_z - C_3^{(i)} q_t^{(i)}, \quad i = 1,2, \tag{3.8}$$

the adsorption rate equations

$$q_t^{(i)} = \frac{k^{(i)}}{c^{(1)} + c^{(2)} + D^{(i)}} (c^{(i)} - q^{(i)}), \quad i = 1,2, \tag{3.9}$$

and the gas phase velocity equation

$$v(t,z) = -C_4 (c_z^{(1)}(t,z) + c_z^{(2)}(t,z)). \tag{3.10}$$

Note that the equations are autonomous, but the boundary conditions are periodic in time. In fact, at the feed end (at $z = 0$) of the adsorber there are two sets of alternating boundary conditions. During the first step, the so-called *pressurization step*, the boundary conditions are

$$\begin{cases} c_z^{(i)}(t,0) = C_1^{-1} C_2 v(t,0)(c^{(i)}(t,0) - y_f^{(i)}), \quad i = 1,2 \\ c^{(1)}(t,0) + c^{(2)}(t,0) = 1 \end{cases} \quad \text{for } 0 \le t \,(\text{mod } 1) < t_f. \tag{3.11}$$

Table 3.1: The dimensionless parameter values for the RPSA process

C_1	0.0042	C_2	1.4134
$C_3^{(1)}$	3.8988	$C_3^{(2)}$	8.3975
C_4	1.8470	C_5	0.0072
$k^{(1)}$	438.2138	$k^{(2)}$	203.4564
$D^{(1)}$	0.5351	$D^{(2)}$	0.5007
$y_f^{(1)}$	0.21	$y_f^{(2)}$	0.79
p_w	0.4717	t_f	0.5

During the second step, the so-called *depressurization step*, the boundary conditions are

$$\begin{cases} c_z^{(i)}(t,0) = 0, \quad i = 1,2 \\ c^{(1)}(t,0) + c^{(2)}(t,0) = p_w \end{cases} \quad \text{for } t_f \leq t \,(\text{mod } 1) \leq 1. \tag{3.12}$$

In this formulation we have rescaled the cycle duration to become one time unit. The boundary conditions at the product end (at $z = 1$) of the adsorber are given by

$$\begin{cases} c_z^{(i)}(t,1) = 0, \quad i = 1,2 \\ v(t,1) = C_5(c^{(1)}(t,1) + c^{(2)}(t,1))^{-1} \end{cases} \tag{3.13}$$

The equations (3.8)-(3.10) together with the boundary conditions (3.11)-(3.13) complete the description of the RPSA model. A solution of (3.8)-(3.10) together with the boundary conditions (3.11)-(3.13) that satisfies

$$\begin{cases} c^{(i)}(0,z) = c^{(i)}(1,z) \\ q^{(i)}(0,z) = q^{(i)}(1,z) \end{cases} \quad i = 1,2,$$

is called a periodic solution of the RPSA model.

The values of the dimensionless parameters are given in Table 3.1. The values of the physical parameters are given in Table A.1, and the relations between the physical parameters and the dimensionless parameters are given in Section A.1.

For the numerical integration of the model equations (3.8)-(3.10), we substitute the equation (3.10) into the equations (3.8), so that we obtain a system of four coupled partial differential equations: two equations for $c^{(i)}(t,z)$, $i = 1,2$ and two equations for $q^{(i)}(t,z)$, $i = 1,2$.

3.3.2 The Cooled Reverse Flow Reactor

A reverse flow reactor (RFR) is a packed bed reactor in which the flow direction is periodically reversed to trap a hot zone within the reactor (see Fig. 3.2). In this way

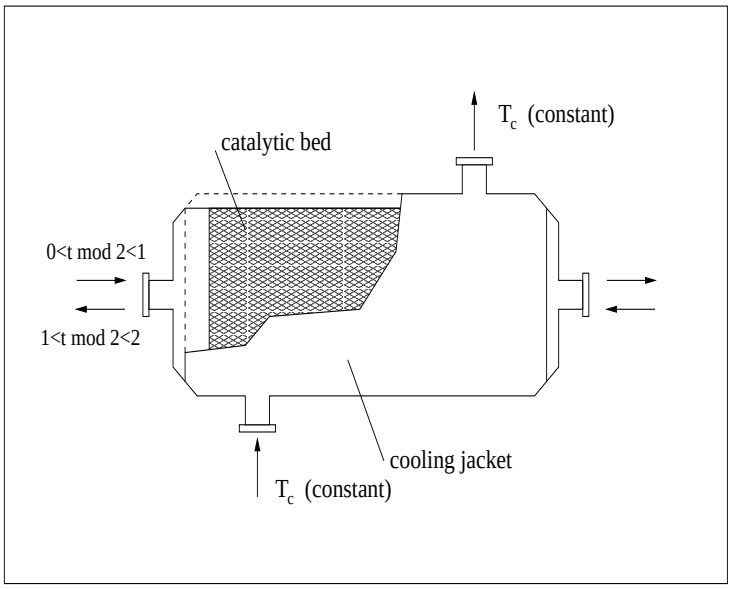

Figure 3.2: Cooled reverse flow reactor.

exothermic reactions can be operated without preheating the feed stream. We consider a cooled RFR in which a single irreversible, exothermic, first order reaction occurs. We describe the RFR by a one-dimensional, pseudo-homogeneous model that accounts for axial heat and mass dispersion and external mass transfer resistance between the fluid and the catalyst. The basic model assumption is to assume that all the physical properties are independent of the temperature and the concentration.

The variables in the model are the dimensionless temperature $\theta : [0,\infty) \times [0,1] \to [0,\infty)$, $(t,z) \to \theta(t,z)$ and the conversion $\chi : [0,\infty) \times [0,1] \to [0,1]$, $(t,z) \to \chi(t,z)$. The dimensionless parameters that appear in the model equations are denoted by K_j for $j = 1,..,7$ and by $g(\theta(t,z))$. The model specific values are given in Table 3.2. For flow from left to right the dimensionless energy and species balances read

$$\theta_t = K_1\theta_{zz} - K_2\theta_z + K_3 g(\theta)(1-\chi) + K_4(1-\theta), \tag{3.14}$$

$$\chi_t = K_5\chi_{zz} - K_6\chi_z + K_7 g(\theta)(1-\chi). \tag{3.15}$$

with boundary conditions

$$\begin{array}{cc} K_1\theta_z(t,0) = K_2(\theta(t,0)-1), & K_5\chi_z(t,0) = K_6\chi(t,0), \\ \theta_z(t,1) = 0, & \chi_z(t,1) = 0. \end{array} \tag{3.16}$$

At each integer value of t the flow direction reverses and the evolution equations and the boundary conditions change accordingly. A *cycle* consists of two flow reversals. This means that one cycle has a duration of two time units. The equations (3.14)-(3.15) with the boundary conditions (3.16) describe the model.

A periodic state of (3.14)-(3.15) with the boundary conditions (3.16) is a solution that

Table 3.2: The dimensionless parameter values for the reverse flow reactor

K_1	$6.9393{\cdot}10^{-4}$	K_2	0.1749
K_3	$1.5577{\cdot}10^{-6}$	K_4	varied (0-0.08)
K_5	$2.4038{\cdot}10^{-3}$	K_6	174.06
K_7	0.01	$g(\theta)$	$\dfrac{1.6656 \cdot 10^{-5} e^{25.785(\theta-1)/\theta}}{1.6656 \cdot 10^{-5} + e^{-25.785/\theta}}$

satisfies

$$\theta(0,z) = \theta(1,1-z),$$
$$\chi(0,z) = \chi(1,1-z).$$

Thus a periodic solution consist of two symmetric parts of one time unit. The values of the dimensionless parameters are given in Table 3.2. The values of the physical parameters are given in Table A.2, and the relations between the physical parameters and the dimensionless parameters are given in Section A.2.

3.4 Computation of the Periodic States

In this section the Newton-Picard method is used to compute periodic states of the rapid pressure swing adsorption process and of the reverse flow reactor. All the computations presented in this section were executed on a Linux desktop computer.

Since the value $F(x)$ is obtained by integrating a large system of differential equations over a period of time with initial condition x, the evaluation of F is a computationally expensive task. This means that for any method used to find a fixed point of F, most of the computation time will be spent in the evaluation of F. Thus the method that needs the least number of evaluations of F will be the most efficient method. Therefore we measure the efficiency of the Newton-Picard method in terms of the number of evaluations of F needed for the computation of a periodic state.

3.4.1 The Rapid Pressure Swing Adsorber

For the rapid pressure swing adsorber we chose to discretize the spatial variable on a grid of 100 nodes using a finite volume approach with second order centered finite differences. After the spatial discretization of the two mass balances and the two adsorption rate equations, a system of 400 ordinary differential equations is obtained and hence $N = 400$. The system of ODE's is integrated in time using the NAG fortran library routine D02EJF. For the δ_0 in the convergence criterion in the Newton-Picard method we take $\delta_0 = 10^{-9}$.

In order to show that the rapid pressure swing adsorber exhibits low dimensional dynamics, in Figure 3.3 a typical plot of the eigenvalues of the Jacobian is depicted. We

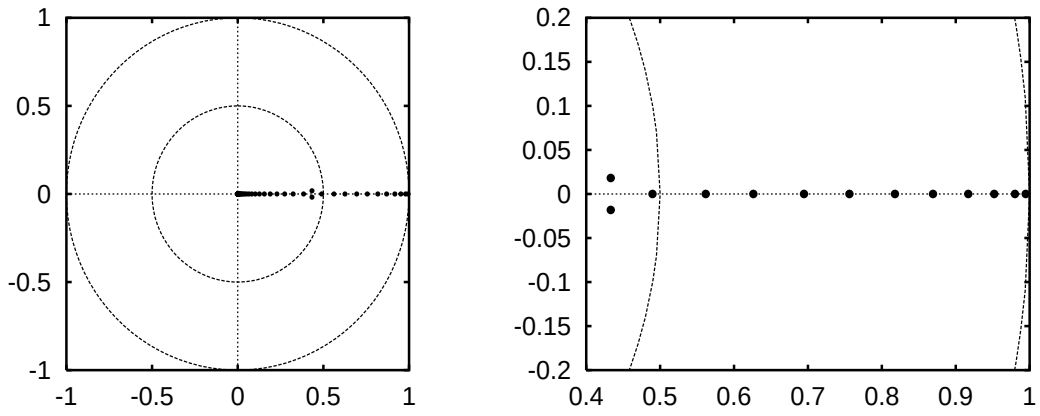

Figure 3.3: The eigenvalues of the Jacobian $J(x^*)$ for the rapid pressure swing adsorber at the periodic state x^* of the (discretized) infinite dimensional period map F plotted in the complex plane. In the right figure we zoom in on the eigenvalues close to 1.

Table 3.3: The number of evaluations of F for the different values of ρ for the rapid pressure swing adsorber. For Newton's method also the number of iterations is given.

	$\rho = 1$ (Picard)	$\rho = 0.75$	$\rho = 0.5$	$\rho = 0.25$	$\rho = 0$ (Newton)
# eval. F	3278	666	560	550	2009 (5 it.)

see that there are only 10 eigenvalues out of the computed 400, that are outside the disc with radius 0.5.

We would like to determine a periodic state for the parameter values given in Table 3.1. The initial conditions for the computations are given by

$$c^{(i)}(0,z) = q^{(i)}(0,z) = y_f^{(i)} p_w.$$

The optimal choice of the parameter ρ in the Newton-Picard method depends on the problem under consideration. There are no good rules for making the choice, except a rule of thumb gathered from experience: in [37] it is reported that values of ρ around 0.5 result in good performance of the method.

We applied the Newton-Picard method for several values of ρ. The number of evaluations of F needed for the computation of the periodic state for these values of ρ are listed in Table 3.3. In Table 3.4 for each value of ρ the corresponding dimension of the subspace U is given. We see that for ρ equal to 0.5 and 0.25 the method is most efficient, which is in agreement with the experiences in [37]. The computed periodic state is depicted in Figure 3.4.

Table 3.4: The dimension of U for each values of ρ for the rapid pressure swing adsorber.

	$\rho = 0.75$	$\rho = 0.5$	$\rho = 0.25$
$\dim(U)$	6	10	14

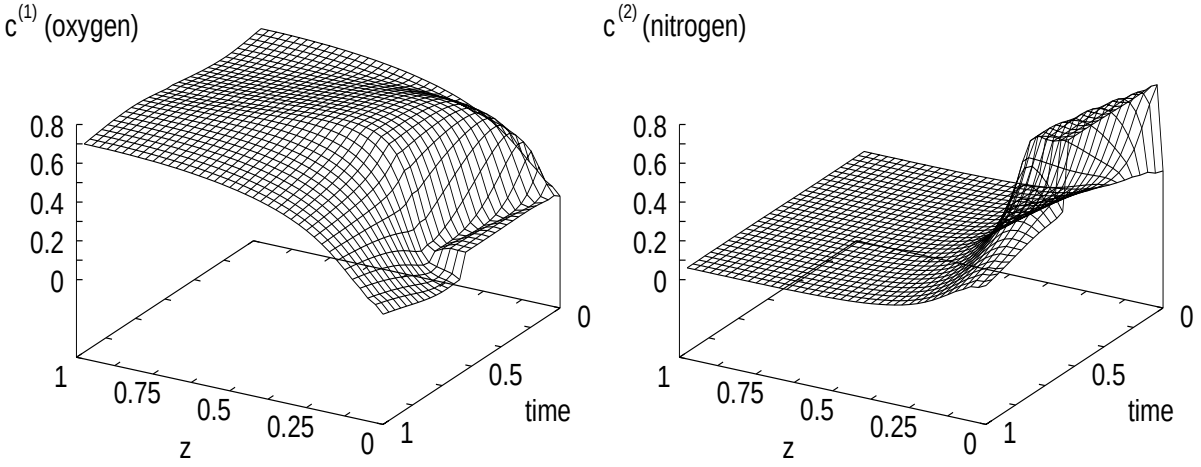

Figure 3.4: The computed periodic state for the rapid pressure swing adsorber: the dimensionless gas phase concentration of oxygen (left) and nitrogen (right) versus the axial distance z and time.

3.4.2 The Cooled Reverse Flow Reactor

For the reverse flow reactor we chose to discretize the spatial variable on a grid of 120 nodes using a finite volume approach with upwinding of the first space derivatives in the mass balance . The discretization of the model equations results for the RFR in a system of 240 ordinary differential equations. Hence $N = 240$. The system of ODE's is integrated in time using the NAG fortran library routine D02EJF. The δ_0 in the Newton-Picard method is taken to be 10^{-7}.

In order to show that the reverse flow reactor also exhibits low dimensional dynamics, in Figure 3.6 a typical plot of the eigenvalues of the Jacobian is depicted. We see that there are only 9 eigenvalues out of the computed 240, that are outside the disc with radius 0.5.

We applied the Newton-Picard method to compute a periodic state of the reverse flow reactor for two different values of the parameter K_4, i.e. $K_4 = 0$ and $K_4 = 0.0175$. The parameter K_4 is a dimensionless measure for the cooling capacity, with $K_4 = 0$ corresponding to no cooling. For both computations the initial conditions

$$\theta(0, z) = 2.02 \text{ and } \chi(0, z) = 0.7,$$

were used. In Table 3.5 the number of function evaluations for the different values of ρ are listed and in Table 3.6 the corresponding dimension of the subspace U is given. We see that for both values of K_4 the method is most efficient with $\rho = 0.5$. For $K_4 = 0.0175$, however, the differences between the various computations is small, and even the Picard iteration ($\rho = 1$) performs well.

As the Newton-Picard method is designed to perform especially well in a continuation context we did not only compute single periodic states for the reverse flow reactor, but

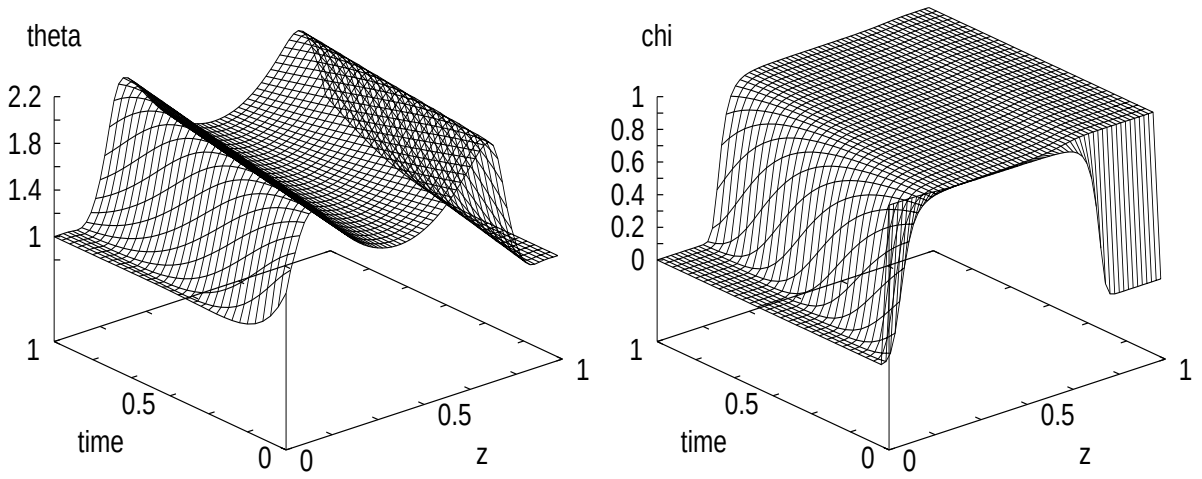

Figure 3.5: The computed periodic state for the reverse flow reactor with the dimensionless cooling capacity $K_4 = 0.0175$: the dimensionless temperature θ (left) and the conversion χ (right) versus the axial distance z and time.

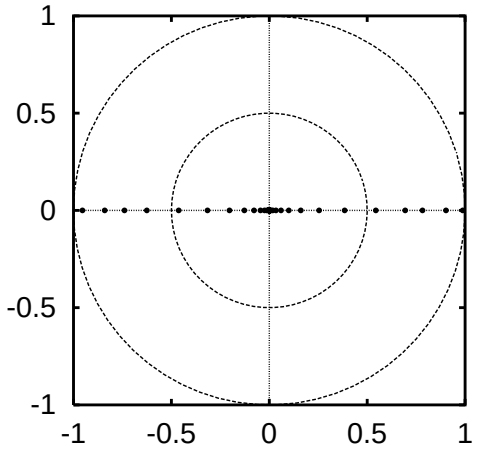

Figure 3.6: The eigenvalues of the Jacobian $J(x^*)$ for the reverse flow reactor with $K_4 = 0$ at the periodic state x^* of the (discretized) infinite dimensional period map F plotted in the complex plane.

Table 3.5: The number of evaluations of F for the different values of ρ for the reverse flow reactor. For Newton's method also the number of iterations is given

	$\rho = 1$ (Picard)	$\rho = 0.75$	$\rho = 0.5$	$\rho = 0.25$	$\rho = 0$ (Newton)
# eval. for $K_4 = 0$	991	396	337	365	1211 (5 it.)
# eval. for $K_4 = 0.0175$	346	376	324	331	729 (3 it.)

Table 3.6: The dimension of U for each values of ρ for the reverse flow reactor.

	$\rho = 0.75$	$\rho = 0.5$	$\rho = 0.25$
$\dim(U)$ for $K_4 = 0$	6	9	13
$\dim(U)$ for $K_4 = 0.0175$	6	9	13

Table 3.7: The number of function evaluations for the different runs of the parameter continuation for the reverse flow reactor.

	$\dim(U)$=6	$\dim(U)$=9	$\dim(U)$=12
# eval.	9335	6408	6853

also a branch of periodic states for a range of values of the parameter K_4. The parameter K_4 was varied in 81 steps from 0 to 0.08. Only for the first periodic state (for $K_4 = 0$) the Jacobian of F and the basis V_p of the subspace U was computed. After that the basis V_p at the end of the previous value of K_4 is used as an approximation of the basis at the new value of K_4. As initial condition the state computed at the previous vale of K_4 was used. For each value of K_4 also the three eigenvalues of J with the largest absolute value were computed with four digits of accuracy. For this purpose after the Newton-Picard iterations in most cases only one subspace iteration was sufficient.

In Figure 3.7 the branch of periodic states is depicted. In the left figure the maximum temperature in the bed at flow reversal is plotted versus K_4. In the right figure for each value of K_4 the eigenvalues with largest absolute value are plotted.

We computed the branch of periodic states with three different choices for the dimension of U. The number of F-evaluations used in these three computations is given in Table 3.7. The run with dimension 9 for U is the most efficient (this value corresponds to $\rho = 0.5$ for the first iteration). For this run the average number of evaluations of F needed for each value of K_4 is less than 80. This is about one third of the computational effort of one Newton iteration ($N = 240$).

The branch of periodic states depicted in Fig. 3.7 is only a part of a longer branch of periodic states. We will compute this longer branch in Chapter 5, see Fig. 5.4. In order to show that it is justified to keep the dimension of the subspace U fixed during the computation of the branch of periodic states, we computed the largest 12 eigenvalues along the branch. The absolute values of these eigenvalues is plotted in Fig. 3.8. We see that the smaller eigenvalues, which are all real, do not change much along the branch.

3.5 Notes

The contents of this chapter are based on [59].

The Newton-Picard method is developed by Lust et al. in [37]. In this paper it is used to compute periodic solutions of large systems of autonomous ordinary differential equations, autonomous partial differential equations and delay differential equations. Methods that are based on similar ideas as the Newton-Picard method, are the adaptive

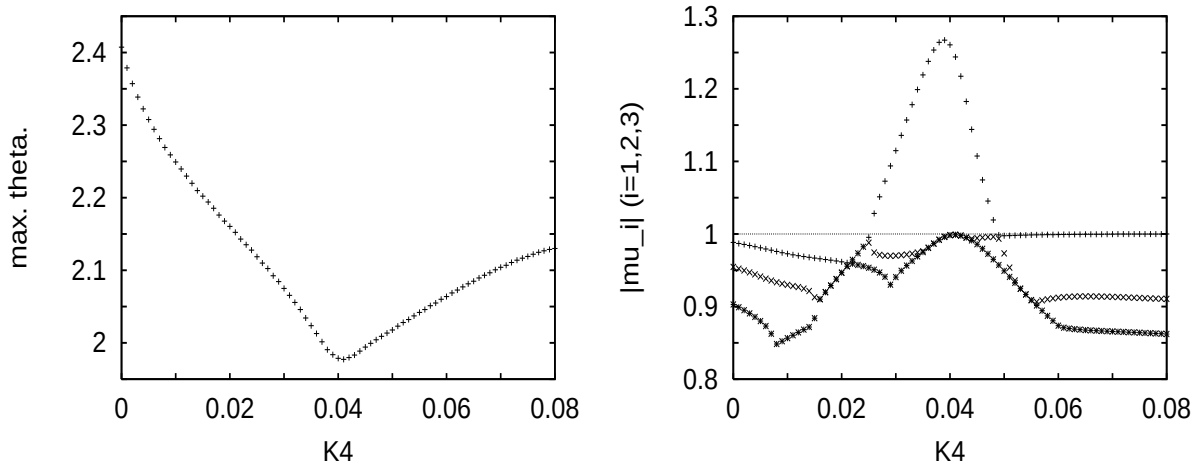

Figure 3.7: The computed branch of periodic solutions for the reverse flow reactor: the maximum temperature in the bed at flow reversal (left) and the moduli of the three eigenvalues $|\mu_i|$ $(i = 1, 2, 3)$ of the Jacobian J at the periodic state with the largest absolute values (right) versus K_4.

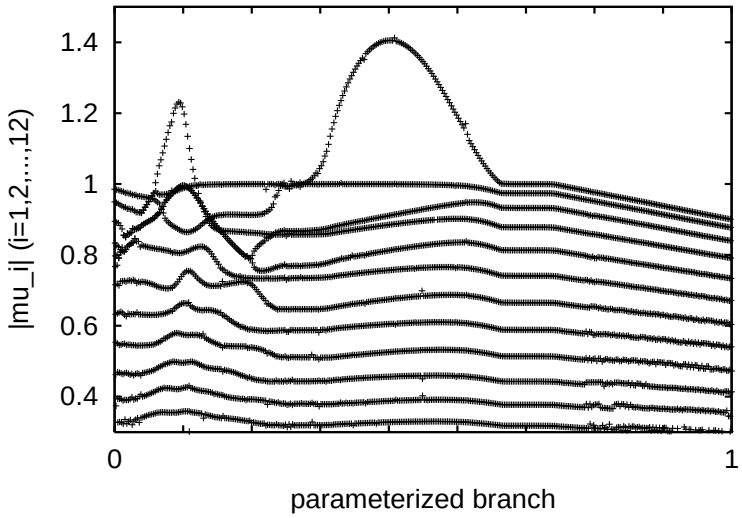

Figure 3.8: The absolute values of the largest eigenvalues of the Jacobian of F along the branch computed in Chapter 5, Fig. 5.4. On the horizontal axis the variable that parameterizes the branch is plotted.

condensation process [19, 20] and the recursive projection method [49].

A good discussion of the subspace iteration algorithm and its convergence properties can be found in [45].

For more details about the convergence analysis of the Newton-Picard method, we refer to [37], where it is demonstrated that the Newton-Picard method converges linearly with rate equal to $|\mu_{p+1}|$.

We start the Newton-Picard method with computing the Jacobian in order to determine p and an initial basis V_p of the subspace U (see Section 3.2). The intimal basis V_p can also be obtained by subspace iteration. Which of the two methods is the least expensive, depends on p, $|\mu_{p+1}|$ and N.

The projection step in the subspace iteration algorithm (see Section 3.2) is strictly speaking not necessary if we keep the dimension of the subspace U fixed during the Newton-Picard iterations. This step is intended for the approach in [37], where the subspace iteration is performed on $p+p_e$ (with $p_e = 2$, 3 or 4) vectors in order to accelerate the convergence of the subspace iteration procedure and to improve the detection of eigenvalues leaving or entering the region $C_\rho := \{z \in \mathbb{C} : |z| < \rho\}$. The projection step aims to find the p vectors, out of the $p + p_e$ dimensional subspace, that span the subspace determined by the p eigenvectors with largest corresponding eigenvalues.

The model equations and the parameter values for the rapid pressure swing adsorber are taken from [38]. The model equations and parameter values for the cooled reverse flow reactor are taken from [21]

The NAG fortran library routine D02EJF that we use for the time integration of the cyclic processes, uses a variable-order, variable-stepsize method implementing the Backward Differentiation Formulae (BDF).

Chapter 4

Comparison of the Newton-Picard Method with Existing Methods

4.1 Introduction

In this chapter we compare the performance of the Newton-Picard method, which we introduced in the previous chapter, with existing methods. These methods are Newton's method, Picard iteration and Broyden's method. We discussed these methods in Section 1.5.

Secondly, we address the question which method to use when. It is not always clear which method will be most appropriate for a given system. This also depends on the type of question asked about the system. In bifurcation analysis, for example, the stability of a computed state is crucial information. Although Broyden's method is efficient in obtaining a periodic state, it does not determine the stability of the state. The Newton-Picard method does compute the eigenvalues that determine the stability.

The organization of this chapter is as follows. In Section 4.2 we discuss two different convergence criteria on which the comparison is based. In Section 4.3 we discuss a variant of the Newton-Picard method that we use in this chapter. In Section 4.4 we give the model equations for a number of cyclic processes and we compare the performance of the Newton-Picard method to the performance of dynamic simulation, Newton's method and Broyden's method, for these example processes. In Section 4.5, we discuss, depending on the characteristics of the chemical process, which acceleration method has the best performance. For linear systems there is a complete answer.

4.2 Method Independent Convergence Criteria

In order to be able to compare two iterative methods, we need a convergence criterion that is independent of the methods. In this chapter, we consider two natural choices for such a convergence criterion.

The first criterion is based on the notion of the *residual* ε_x of a state of the system x,

defined by

$$\varepsilon_x := |F(x) - x|. \tag{4.1}$$

We say that a method has reached a periodic state after i iterations, if the residual ε_{x_i} satisfies

$$\varepsilon_{x_i} < \delta_0 \ , \tag{4.2}$$

where δ_0 only depends on the chemical process and on the required precision.

The second criterion in based on the *deviation* γ_x of a state x which is defined by

$$\gamma_x := |x^* - x|, \tag{4.3}$$

where x^* denotes the true periodic solution that we would like to compute. We again say that a method has reached a periodic state after i iterations, if the deviation γ_{x_i} satisfies

$$\gamma_{x_i} < \delta_0^* \ , \tag{4.4}$$

where δ_0^* depends on the system and on the required precision. The weak point of this latter convergence criterion is that it is based on the true periodic state. This can be resolved by first precomputing a very accurate approximation of the true periodic state and then by using this approximation as the true periodic state. This makes that this criterion is only of theoretical interest, since in practice it does not make sense to presolve for the "true" solution.

Now it is relevant to observe whether or not the choice of the convergence criterion influences the conclusions of the comparison. Therefore we assessed the computations for the different chemical systems using both convergence criteria.

4.3 Newton-Picard Revisited

In this chapter we also experiment with a slight adaptation of the Newton-Picard method. The adaptation is intended to make a further reduction of the number of evaluations of the period map. The idea is to keep the subspace U fixed when the residual becomes small. This results in slight modifications in **Step 6** and **Step 10**, so that the the Newton-Picard algorithm becomes

Newton-Picard Algorithm

Step 1: *Choose an initial value x_0 and set $i = 0$.*

Step 2: *Compute the Jacobian $J(x_0)$ of F at x_0 (finite differences: $N+1$ F-evaluations).*

Step 3: *Compute the eigenvalues and eigenvectors of $J(x_0)$. This determines p, the dimension of the slow subspace U and an orthonormal basis V_{p_0} of U.*

Step 4: $x_1 = x_0 - (J(x_0) - I)^{-1}(F(x_0) - x_0)$.

Step 5: *If* $|F(x_i) - x_i| < \delta_0$ *then stop (1 evaluation of* F*).*

Step 6: *If* $|F(x_i) - x_i| > \delta_1$ *then set j=1 and iterate s times: (sp evaluations of* F*).*

1) *compute the ordered real Schur-factorization* $Y^T RY$ *of* $V_{p_i}^T S$*, with* $S = J(x_i)V_{p_i}$.

2) *if* $j < s$ *then: orthonormalize* $V = SY$ *and put the result in* V_{p_i}*. Set* $j = j + 1$.

Step 7: $v_0 = (I - V_{p_i}V_{p_i}^T)(F(x_i) - x_i)$*. Iterate* $v_{j+1} = (I - V_{p_i}V_{p_i}^T)(J(x_i)v_j + (F(x_i) - x_i))$ *l times. Set* $V_r\Delta\bar{r} := v_l$*. (Computation of* $J(x_i)v_j$*: l evaluations of* F*).*

Step 8: *Solve* $(V_{p_i}^T S - I)(\Delta\bar{p}) + V_{p_i}^T J(x_i)(V_r\Delta\bar{r}) = -V_{p_i}^T(F(x_i) - x_i)$ *for* $\Delta\bar{p}$*. (Computation of* $J(x_i)(V_r\Delta\bar{r})$*: 1 evaluation of* F*).*

Step 9: *Set* $x_{i+1} := x_i + V_{p_i}\Delta\bar{p} + V_r\Delta\bar{r}$.

Step 10: *If* $|F(x_i) - x_i| > \delta_1$ *then compute* $V = SY$*, orthonormalize* V *and put the result in* $V_{p_{i+1}}$*. Set* $i = i + 1$*. Go to* **Step 5**.

The δ_1 is chosen such that $\delta_1 > \delta_0$. Just like the version of the Newton-Picard algorithm in the previous chapter, the above algorithm amounts to $sp + 2 + l$ evaluations of F per iteration, as long as $|F(x_i) - x_i| \geq \delta_1$. When $|F(x_i) - x_i| < \delta_1$, then the algorithm needs only $2 + l$ evaluations of F per iteration. For the example systems in this chapter we take $l = 2$ and $s = 2$. Thus, for each iteration of the Newton-Picard method, we need $2p + 4$ evaluations of F and when $|F(x_i) - x_i| < \delta_1$ only 4.

4.4 Comparison of the Methods

In this section, we apply the Newton-Picard method to compute periodic states of several different cyclic systems. The performance of the method is compared to the performance of dynamic simulation, Broyden's method and Newton's method. All computations were carried out on an IBM RS6000 SP2 platform.

4.4.1 CO_2/N_2 Separation

The first system is a pressure swing adsorber used to remove CO_2 from N_2 using 5A zeolite. CO_2 is the only component to adsorb. The bed is assumed to be operated adiabatically and account is made for axial temperature gradients and adsorption rate is modeled using the linear driving force model. The composition of the feed gas is 10% CO_2 and 90% N_2.

Model and Parameters for the CO_2/N_2 PSA System

In this section we give the equations that model the CO_2/N_2 PSA system. The variables in the model are the dimensionless mole fractions of the components in the gas phase, the dimensionless adsorbed amount of both components, the dimensionless gas velocity and the dimensionless temperature. The mole fractions of the components in the gas phase are denoted by $y^{(i)} : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to y^{(i)}(t,z)$ $(i = 1,2)$, where z denotes the dimensionless axial coordinate and t the dimensionless time. The component CO_2 is denoted by $i = 1$ and the component N_2 is denoted by $i = 2$. The component CO_2 is the only component to adsorb and the adsorbed amount of this component in the solid phase is denoted by $q^{(1)} : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to q^{(1)}(t,z)$. The velocity of the gas phase is denoted by $v : [0,\infty) \times [0,1] \to \mathbb{R},\ (t,z) \to v(t,z)$. The temperature is denoted by $\theta : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to \theta(t,z)$. The pressure is denoted by $d : [0,\infty) \to [0,\infty),\ t \to d(t)$ and is given in Table 4.1.

The equations that describe the behaviour of these variables for the CO_2/N_2 PSA system can be written as

$$y_t^{(1)} = K_1 y_{zz}^{(1)} - (v y^{(1)})_z - K_2 \frac{\theta}{d} q_t^{(1)} - \frac{y^{(1)}}{d} d_t, \tag{4.5}$$

$$y_t^{(2)} = K_1 y_{zz}^{(2)} - (v y^{(2)})_z - \frac{y^{(2)}}{d} d_t, \tag{4.6}$$

$$q_t^{(1)} = \frac{k}{d}(g^{(1)} - q^{(1)}), \tag{4.7}$$

$$\theta_t = \left[K_4 \theta_{zz} - \frac{C d v}{\theta}\theta_z + K_2 H_1 q_t^{(1)} + K_5(\theta - 1)\right]\left[\frac{Cd}{\theta} + K_3\right]^{-1}, \tag{4.8}$$

where v is given by

$$v(t,z) = -\frac{K_2 \theta(t,z)}{d(t)} \int_a^z q_t^{(1)}(t,z)dz - \frac{z-a}{d(t)} d_t(t) + v(t,a), \tag{4.9}$$

where $v(t,a)$ is given as a boundary condition (with a being either 0 or 1, depending on the end of the reactor where the boundary condition is given, see Table 4.9). The equilibrium concentrations $g^{(1)}$ is given by

$$g^{(1)}(\theta, d, y^{(1)}) = \frac{\kappa_1 \kappa_2 e^{\kappa_3/\theta} d y^{(1)} \theta^{\alpha-1}}{1 + \kappa_2 e^{\kappa_3/\theta} d y^{(1)} \theta^{\alpha-1}}.$$

The boundary conditions and the pressure differ from step to step and are given in the Tables 4.1-4.3. The physical parameters for the CO_2/N_2 PSA system are given in Section A.3, as well as their relations to the dimensionless parameters. Initial conditions are given in Table 4.4.

The composition of the purge gas is equal to the averaged composition of the product gas obtained in the adsorption step. This feedback is expressed in the formula

$$y_{\mathrm{p}}^{(i)} = \int_0^{t_{\mathrm{ads}}/T} \frac{v(t,1)d(t)}{\theta(t,1)} y^{(i)}(t,1)dt \left[\int_0^{t_{\mathrm{ads}}/T} \frac{v(t,1)d(t)}{\theta(t,1)} dt\right]^{-1}.$$

Table 4.1: The periodic values of the pressure $d(t)$ and the boundary conditions for the velocity $v(t,z)$ for the CO_2/N_2 and H_2S/natural gas PSA systems

step	v (both systems)	$d(t)$ (CO_2/N_2)	$d(t)$ (H_2S/nat. gas)
pres.	$v(t,1)=0$	$\frac{P_L}{P_H}+\frac{Tt}{t_{pre}}\left(1-\frac{P_L}{P_H}\right)$	$\frac{P_L}{P_H}+\frac{Tt}{t_{pre}}\left(1-\frac{P_L}{P_H}\right)$
ads.	$v(t,0)=v_{ads}/U$	1	1
blow.	$v(t,1)=0$	$1-\left(1-\frac{P_L}{P_H}\right)\frac{Tt}{t_{blo}}$	$\frac{P_L}{P_H}+(1-\frac{P_L}{P_H})e^{-Tt/t_s}$
pur.	$v(t,1)=v_{pur}/U$	$\frac{P_L}{P_H}$	$\frac{P_L}{P_H}$

Table 4.2: The boundary conditions for $y^{(i)}$ for the CO_2/N_2 and H_2S/natural gas PSA systems

step		
pres.	$K_1 y_z^{(i)}(0,t)=v(t,0)(y^{(i)}(0,t)-y_f^{(i)})$	$y_z^{(i)}(t,1)=0$
ads.	$K_1 y_z^{(i)}(0,t)=v(t,0)(y^{(i)}(0,t)-y_f^{(i)})$	$y_z^{(i)}(t,1)=0$
blow.	$y_z^{(i)}(t,0)=0$	$y_z^{(i)}(t,1)=0$
pur.	$y_z^{(i)}(0,t)=0$	$K_1 y_z^{(i)}(t,1)=v(t,1)(y^{(i)}(t,1)-y_p^{(i)})$

Note that $y^{(1)}$ and $y^{(2)}$ denote mole fractions, so that $y^{(1)}(t,z)+y^{(2)}(t,z)=1$. For the numerical integration of the model equations (4.5)-(4.8), we use this relation to eliminate the equation for $y^{(2)}$. We also substitute the equation (4.9) into the equation (4.5) and (4.8). In this way we obtain for the numerical integration a system of three coupled partial differential equations for $y^{(1)}(t,z)$, $q^{(1)}(t,z)$ and $\theta(t,z)$.

Results

In order to produce results with a moderate accuracy within a reasonable computation time, the spatial variable z is discretized on a grid of 40 equidistant nodes using first order upwind finite differences. The resulting system of 120 (temperature 40 nodes, gas phase concentration 40 nodes, solid phase concentration 40 nodes) ordinary differential equations is integrated using the trapezoidal rule with variable step size. Hence, $N=120$.

The results for the CO_2/N_2 PSA system with $\delta_0=10^{-10}$ and $\delta_0^*=10^{-8}$ are given in Table 4.5. For each method and convergence criterion, the number of cycle simulations is listed. It is seen that the results differ for the two convergence criteria, the difference being the largest for Newton's method. However, the ordering of the methods from the most efficient to the least efficient (i.e. Broyden, dynamic simulation, Newton-Picard, Newton) is for both convergence criteria the same.

The optimal choice of the parameter ρ in the Newton-Picard method depends on the

Table 4.3: The boundary conditions for the dimensionless temperature θ for the CO_2/N_2 and H_2S/natural gas PSA systems

step		
pres.	$K_5\theta_z(t,0) = \frac{Cd(t)}{\theta(t,0)}v(t,0)(\theta(t,0)-1)$	$\theta_z(t,1) = 0$
ads.	$K_5\theta_z(t,0) = \frac{Cd(t)}{\theta(t,0)}v(t,0)(\theta(t,0)-1)$	$\theta_z(t,1) = 0$
blow.	$\theta_z(t,0) = 0$	$\theta_z(t,1) = 0$
pur.	$\theta_z(t,0) = 0$	$K_5\theta_z(t,1) = \frac{Cd(t)}{\theta(t,1)}v(t,1)(\theta(t,1)-1)$

Table 4.4: The initial conditions for the CO_2/N_2 PSA system

saturated bed	empty bed
$y^{(1)}(0,z) = y_{\mathrm{f}}^{(1)}$	$y^{(1)}(0,z) = 0.01 \cdot y_{\mathrm{f}}^{(1)}$
$y^{(2)}(0,z) = 1 - y^{(1)}(0,z)$	$y^{(2)}(0,z) = 1 - y^{(1)}(0,z)$
$q^{(1)}(0,z) = g_{\mathrm{C}}^{(i)}(\theta(0,z), d(0), y^{(1)}(0,z))$	$q^{(1)}(0,z) = g_{\mathrm{C}}^{(i)}(\theta(0,z), d(0), y^{(1)}(0,z))$
$\theta(0,z) = 1$	$\theta(0,z) = 1$
$d(0) = 1$	$d(0) = 1$

Table 4.5: The number of cycle simulations for each method. The first number is obtained using the residual convergence criterion and the second number using the deviation convergence criterion. [1]: computation started with initial condition an empty bed. [2]: computation started with initial condition a saturated bed. [3]: computation for $T = 0.2$. [4]: computation for $T = 0.3$.

	Dyn. Sim.	Newton	Broyden	New.-Pic.
H_2O/air	769/708	515/514	99/84	-
CO_2/He	31/22	515/514	14/11	-
CO_2/N_2	200/155	857/494	33/31	216/196
H_2S/nat. gas[1]	579/468	1452/605	125/104	225/213
H_2S/nat. gas[2]	700/613	>2000/968	297/268	375/355
RFR[3]	-	-	58/53	322/302
RFR[4]	-	-	44/40	234/230

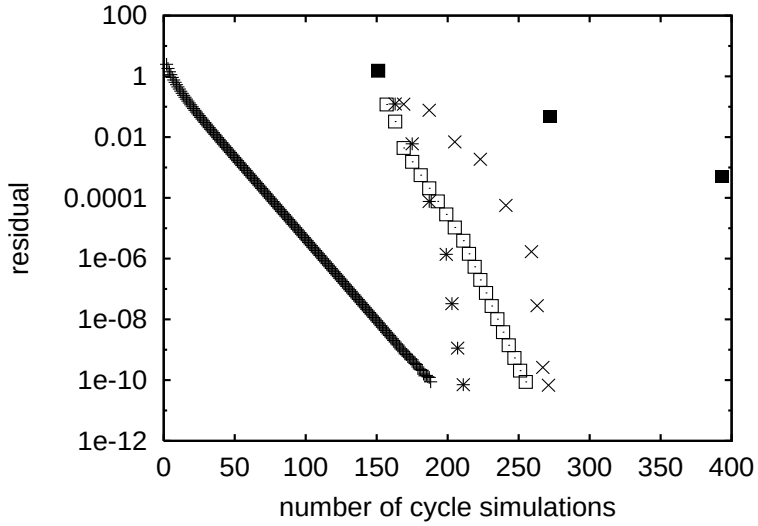

Figure 4.1: The residual versus the number of cycle simulations for the Newton-Picard method applied to the CO_2/N_2 PSA separation for different values of ρ: $+$: $\rho = 1$ (dynamic simulation); $\times$: $\rho = 0.2$; $*$: $\rho = 0.5$; $\square$: $\rho = 0.8$; $\blacksquare$: $\rho = 0$ (For Newton's method, only the first three iterations are shown).

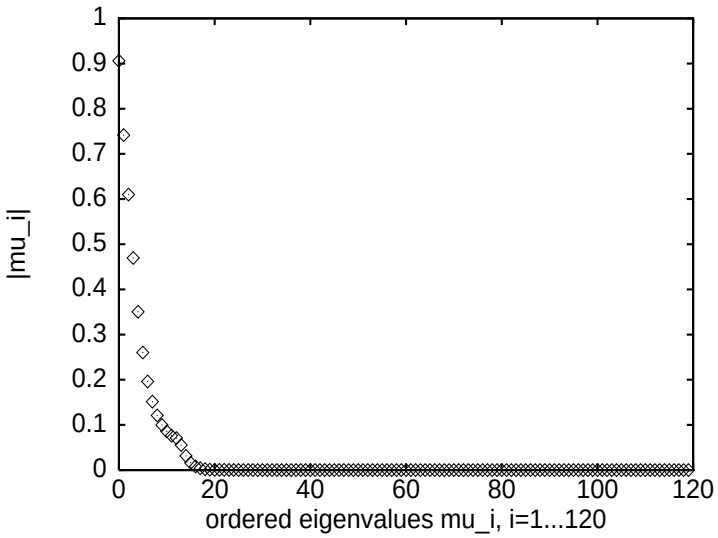

Figure 4.2: The modulus of the eigenvalues of the Jacobian for the CO_2/N_2 separation calculated for an initial state corresponding to a saturated bed.

problem under consideration. Remember that the parameter ρ has to be a real number ranging from 0 to 1 and determines the dimension of U. In the extreme cases $\rho = 1$ and $\rho = 0$, the Newton-Picard method reduces, respectively, to dynamic simulation and to Newton's method.

There are no good rules for choosing ρ, except a rule of thumb gathered from experience: in [37] it is reported that values of ρ around 0.5 result in good performance of the method. Our computations for the CO_2/N_2 PSA system confirm this value of 0.5 for ρ. In Fig. 4.1, the residual is plotted versus the number of cycle simulations using the Newton-Picard method with different values of ρ. In this specific example, it is seen that the computation for $\rho = 1$ (dynamic simulation) is most efficient, but this is due to that for this value of ρ in the first iteration the Jacobian does not need to be computed. Excluding the dynamic simulation, i.e., $\rho = 1$, the computation with $\rho = 0.5$ is most efficient.

Therefore we used $\rho = 0.5$ for the Newton-Picard computations shown in Table 4.5. We will also see that the value of ρ influences the robustness of the method. The parameter δ_1, which influences whether an update of the Jacobian on the slow subspace is made, is taken to be 10^{-5} (obtained as an "trial and error" optimum).

From Fig. 4.2, it is seen that only three eigenvalues exist with modulus greater than 0.5. Thus the "slowly converging" subspace U is three dimensional. This means that for each iteration of the Newton-Picard method only 10 cycles (and when the residual is smaller than 10^{-5} only 4 cycles) need to be simulated instead of the 120 for the full Newton scheme. Thus the Newton-Picard iterations require very little computational effort. Only the first iteration, when the Jacobian is calculated, needs more time. However, from Table 4.5, we conclude that also Broyden's method benefits from this situation and, actually, Broyden's method turns out to be much faster than he Newton-Picard method. From this, one can identify the major drawback of the Newton-Picard method: it needs to calculate the slowly converging subspace, before it can make the first iteration. In our implementation the determination of the slow subspace needs $N + 1$ cycle simulations. One should notice, however, that the Newton-Picard method does also give the stability properties of the computed periodic solution.

The computations shown in Table 4.5 all started from a saturated bed (see Table 4.4). For nonlinear systems, it can be expected that the performance of the convergence acceleration methods depends on the initial conditions. For the CO_2/N_2 system, however, it turns out that when the computations are started from an empty bed, results similar to those in Table 4.5 are obtained. Thus, for this PSA system, the convergence of the different methods does not depend much on the initial conditions. For the PSA system considered in the next section, we do find that the performance of the method does depend on the initial conditions.

4.4.2 H_2S/Natural Gas Separation

The second system is a pressure swing adsorber used to remove H_2S from natural gas using 5A zeolite. The model natural gas consists of 1000 ppm H_2S, 5% CO_2 and 94.9% CH_4. All three components are adsorbed simultaneously. Explicitly the energy balance is included and axial temperature gradients are accounted for. The most important modeling assumptions for this system are that locally the adsorption equilibrium is maintained and that the adsorption isotherm is of the Langmuir type.

Model and Parameters for the H_2S/Natural Gas PSA System

In this section we give the equations that model the H_2S/natural gas PSA system. The variables in the model are the dimensionless mole fractions of the components in the gas phase, the dimensionless adsorbed amount of both components, the dimensionless gas velocity and the dimensionless temperature. The mole fractions of the components in the gas phase are denoted by $y^{(i)} : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to y^{(i)}(t,z),\ (i = 1,2,3)$ where z denotes the dimensionless axial coordinate and t the dimensionless time. The component H_2S is denoted by $i = 1$, the component CO_2 is denoted by $i = 2$ and the component CH_4 is denoted by $i = 3$. The adsorbed amount of the components in the solid phase is denoted by $q^{(i)} : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to y^{(i)}(t,z)$. The velocity of the gas phase is denoted by $v : [0,\infty) \times [0,1] \to \mathbb{R},\ (t,z) \to v(t,z)$. The temperature is denoted by $\theta : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to \theta(t,z)$. The pressure is denoted by $d : [0,\infty) \to [0,\infty),\ t \to d(t)$ and is given in Table 4.1.

The equations that describe the behaviour of these variables for the H_2S/natural gas PSA system can be written as

$$y_t^{(i)} = K_1 y_{zz}^{(i)} - (v y^{(i)})_z - K_2 \frac{\theta}{d} q_t^{(i)} - \frac{y^{(i)}}{d} d_t, \tag{4.10}$$

$$q^{(i)} = g^{(i)}(\theta, d, y^{(1)}, y^{(2)}, y^{(3)}), \tag{4.11}$$

$$\theta_t = \left[K_4 \theta_{zz} - \frac{Cdv}{\theta} \theta_z + K_2 \sum_{i=1}^{3} H_i q_t^{(i)} + K_5(\theta - 1) \right] \left[\frac{Cd}{\theta} + K_3 \right]^{-1}, \tag{4.12}$$

where v is given by

$$v(t,z) = -\frac{K_2 \theta(t,z)}{d(t)} \int_a^z \sum_{i=1}^{3} q_t^{(i)}(t,z) dz - \frac{z-a}{d(t)} d_t(t) + v(t,a), \tag{4.13}$$

where $v(t,a)$ is given as a boundary condition (with a being either 0 or 1, depending on the end of the reactor where the boundary condition is given, see Table 4.9). The equilibrium concentration $g^{(i)}$ is given by

$$g^{(i)}(\theta, d, y^{(1)}, y^{(2)}, y^{(3)}) = \frac{(\hat{k}_1^{(i)} - \hat{k}_2^{(i)} \theta) \hat{k}_3^{(i)} e^{\hat{k}_4^{(i)}/\theta} dy^{(i)}}{1 + \sum_{j=1}^{n} \hat{k}_3^{(j)} e^{\hat{k}_4^{(j)}/\theta} dy^{(j)}}.$$

Table 4.6: The initial conditions for the H_2S/natural gas PSA systems

saturated bed	empty bed
$y^{(1)}(0,z) = y_{\mathrm{f}}^{(1)}$	$y^{(1)}(0,z) = 0.01 \cdot y_{\mathrm{f}}^{(1)}$
$y^{(2)}(0,z) = y_{\mathrm{f}}^{(2)}$	$y^{(2)}(0,z) = y_{\mathrm{f}}^{(2)}$
$y^{(3)}(0,z) = 1 - y^{(1)}(0,z) - y^{(2)}(0,z)$	$y^{(3)}(0,z) = 1 - y^{(1)}(0,z) - y^{(2)}(0,z)$
$q^{(i)}(0,z) = g_{\mathrm{D}}^{(i)}(\theta, d, y^{(1)}, y^{(2)}, y^{(3)})$	$q^{(i)}(0,z) = g_{\mathrm{D}}^{(i)}(\theta, d, y^{(1)}, y^{(2)}, y^{(3)})$
$\theta(0,z) = 1$	$\theta(0,z) = 1$
$d(0) = 1$	$d(0) = 1$

The boundary conditions and the pressure differ from step to step and are given in the Tables 4.1-4.3. The physical parameters for the H_2S/natural gas PSA systems are given in Section A.3, as well as their relations to the dimensionless parameters. Initial conditions are given in Table 4.6.

The composition of the purge gas is equal to the averaged composition of the product gas obtained in the adsorption step. This feedback is expressed in the formula

$$y_{\mathrm{p}}^{(i)} = \int_0^{t_{\mathrm{ads}}/T} \frac{v(t,1)d(t)}{\theta(t,1)} y^{(i)}(t,1)dt \left[\int_0^{t_{\mathrm{ads}}/T} \frac{v(t,1)d(t)}{\theta(t,1)} dt \right]^{-1}.$$

Note again that $y^{(i)}$, $i = 1, 2, 3$, denote mole fractions, so that $y^{(1)}(t,z) + y^{(2)}(t,z) + y^{(3)}(t,z) = 1$. For the numerical integration of the model equations (4.5)-(4.8), we use this relation to eliminate the equation for $y^{(3)}$. We also substitute the equation (4.9) into the equation (4.5) and (4.8) and substitute the equation (4.11) for $q^{(i)}$ into (4.10) and (4.12). In this way we obtain for the numerical integration a system of three coupled partial differential equations for $y^{(1)}(t,z)$, $y^{(2)}(t,z)$ and $\theta(t,z)$.

Results

In order to produce results with a moderate accuracy within a reasonable computation time, first the spatial variable z is discretized on a grid of 40 equidistant nodes using first order upwind finite differences. The resulting system of 120 (two gas phase concentrations each 40 nodes and temperature 40 nodes) ordinary differential equations is integrated using the trapezoidal rule with 800 steps per cycle. Hence, $N = 120$.

As before we use the convergence criterion based on the residual defined by (4.1)-(4.2) with $\delta_0 = 10^{-10}$ and the convergence criterion based on the deviation defined by (4.3)-(4.4) with $\delta_0^* = 10^{-8}$. The parameter δ_1 in the Newton-Picard method, which influences whether an update of the Jacobian on the slow subspace is made, is taken to be 10^{-5}.

Table 4.7: The robustness of Broyden's method for the H_2S/nat. gas system starting from a saturated bed. Each computation was started with a number of dynamic simulation cycles. This number is denoted in the first column. The state of the system after these initial cycles is used as the initial condition for Broyden's method. The figure in the second column denotes the number of Broyden iterates before break-down due to bad initial conditions or before convergence. The break-down or convergence is denoted in the last column. (*: based on residual convergence criterion.)

# dyn. sim. iterations	# Broyden iterations	convergence
5	36	no convergence
50	6	no convergence
100	6	no convergence
150	93	no convergence
200	97*	converged

All methods are started from two different initial states (see Table 4.6). The first initial state is an empty bed and the second is a saturated bed. From Table 4.5, it is seen that the dynamic simulation needs more time to converge to a periodic state, starting from a saturated bed than starting from an empty bed. The asymptotic convergence rate, however, is the same. Newton's method yields no improvement for these initial conditions as compared to the dynamic simulation.

The convergence behaviour of Broyden's method depends on the initial conditions. Starting from an empty bed, Broyden's method performs very well. However, when Broyden's method is started from a saturated bed, after a few iterations the method produces badly posed initial conditions for the cycle simulation, so that the step length of the time integration becomes too small and time integration stops. In order to overcome this problem, the computation can be started with a number of dynamic simulation iterations with the aim to obtain initial conditions closer to the periodic state and, therefore, in the region of attraction for Broyden's method. In Table 4.7, the number of Broyden iterations is given after performing a varying number of dynamic simulation cycles. In the last column of this table, it is indicated whether the computation converges to a periodic state or breaks down due to a bad initial condition for the cycle simulation. Note that 200 cycles have to be simulated before a state is obtained close enough to the periodic state for Broyden's method to converge. The computation that starts with the 200 dynamic simulation cycles is shown in Table 4.5. This computation is still faster than the dynamic simulation.

The Newton-Picard method as presented in Section 4.3 might suffer from a lack of robustness when the dimension of the slow subspace is too small, and this happens, for example, when the computation is started from a saturated bed. In Table 4.8, we show the number of iterations needed for the Newton-Picard method with different values of ρ. It is also indicated whether the calculation converges or breaks down. We have seen that the parameter ρ determines the dimension of the slow subspace. In Figure 4.3, we show that for $\rho = 0.5$, there are 9 eigenvalues with modulus larger than 0.5 so that the dimension of U is 9. For $\rho = 0.2$, 16 eigenvalues with modulus larger than 0.2 exist and

Table 4.8: The robustness of the Newton-Picard method for the H_2S/nat. gas system starting from a saturated bed. Each computation is started with a different value of ρ. This value is denoted in the first column. The value of ρ influences the dimension of the slow subspace U, that is given in the second column. In the third column the number of Newton-Picard iterations before break-down or convergence is given. The break-down or convergence is denoted in the last column. (*: based on residual convergence criterion.)

ρ	dim(U)	# New.-Pic. iterations	convergence
0.5	9	3	no convergence
0.2	16	4	no convergence
0.1	21	8*	converged

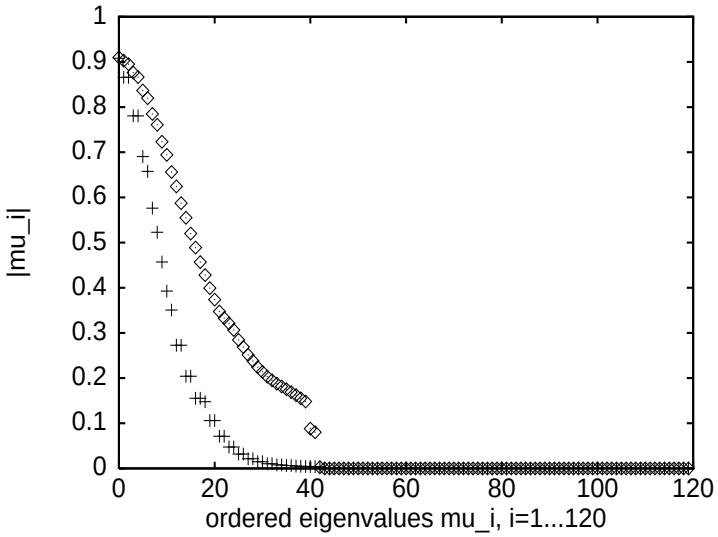

Figure 4.3: The modulus of the eigenvalues of the Jacobian for the H_2S/nat. gas separation calculated at the two different initial states; $\diamond$: empty bed, $+$: saturated bed.

the dimension of U is 16. For $\rho = 0.1$, the dimension of U turns out to be 21. Only in the latter case the Newton-Picard method converges. In particular, it follows the value 0.5 is not always the best choice for the parameter ρ in the Newton-Picard method (see Section 4.4.1).

It is again seen that the results differ for the two convergence criteria, the difference being the largest for Newton's method. However, the ordering of the methods from the most efficient to the least efficient (for both initial conditions: Broyden, Newton-Picard, dynamic simulation, Newton) is for both convergence criteria the same.

4.4.3 Reverse Flow Reactor

The third system is a model of a reverse flow reactor. The model describes a catalytic combustion using monolith type catalysts, and it consists of an energy balance and a mass balance. The model exhibits multiple periodic states which can be either stable or unstable.

Model and Parameters for the Reverse Flow Reactor

In this section we give the equations that model the reverse flow reactor. The variables in the model are the dimensionless temperature rise

$$\theta : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to \theta(t,z),$$

and the conversion

$$\chi : [0,\infty) \times [0,1] \to [0,1],\ (t,z) \to \chi(t,z),$$

where z denotes the axial coordinate and t the time coordinate. The equations that describe the behaviour of θ and χ are given by

$$\begin{aligned} 0 &= \frac{1}{Pe_{\mathrm{c}}}\chi_{zz} - \chi_z + k^*(1-\chi)e^{E^*/(\theta+\theta^*)}, \\ \theta_t &= \frac{1}{Pe_{\mathrm{t}}}\theta_{zz} - \theta_z + k^*(1-\chi)e^{E^*/(\theta+\theta^*)}. \end{aligned}$$

The boundary conditions are given by

$$\begin{aligned} \theta_z(t,0) = Pe_{\mathrm{t}}\theta(t,0), \quad \theta_z(t,1) = 0, \\ \chi_z(t,0) = Pe_{\mathrm{t}}\chi(t,0), \quad \chi_z(t,1) = 0. \end{aligned}$$

The time-periodic boundary conditions are given by

$$\begin{aligned} \theta(1,z) = \theta(0,1-z), \\ \chi(1,z) = \chi(0,1-z). \end{aligned}$$

The physical parameters are given in Section A.5, as well as their relations to the dimensionless parameters.

Results

For the numerical simulation of one cycle of the reverse flow reactor, first the spatial variable z is discretized on a grid of 160 equidistant nodes using first order upwind finite volumes. The resulting system of 160 ordinary differential equations is integrated using the NAG fortran library routine D02EJF. Hence, $N = 160$.

Again, as before, we use the convergence criterion based on the residual ε_x defined by (4.1)-(4.2) with $\delta_0 = 10^{-12}$ and the convergence criterion based on the deviation γ_x defined by (4.3)-(4.4) with $\delta_0^* = 10^{-8}$. The parameter δ_1 in the Newton-Picard method is taken to be 10^{-4}. The dimension of the subspace U is kept fixed at 6 for all the presented computations for the RFR. Computations with different dimensions show that the number of cycle simulations does not depend much on the chosen dimension.

From the previous system, it is seen that the dynamic simulation and Newton's method need more time than the other two methods. Therefore, only Broyden's method and the Newton-Picard method are used for the computation of periodic states of the RFR.

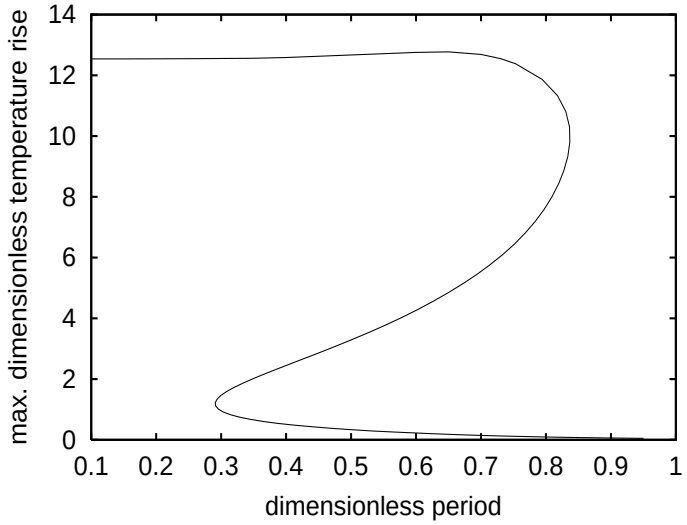

Figure 4.4: A branch of periodic states of the RFR. On the horizontal axis the cycle length is given and on the vertical axis the maximum temperature in the bed before flow reversal is given.

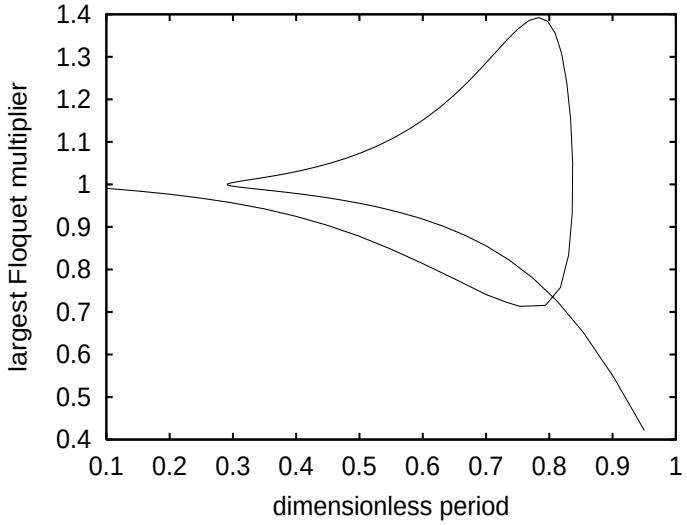

Figure 4.5: A branch of periodic states of the RFR. On the horizontal axis the cycle length is given and on the vertical axis the largest Floquet multiplier is given.

In Table 4.5 results are shown for two different values of the dimensionless period. The computations for the period equal to 0.2 are started with an initial condition a uniform dimensionless temperature equal to 5 throughout the bed. The calculations converge to an ignited state (the maximum dimensionless temperature rise is approximately 12). The computations for the period equal to 0.3 are started with initial condition given by an uniform temperature equal the feed temperature throughout the bed. The calculations converge to an extinguished state (the maximum dimensionless temperature rise is approximately 1). In both cases, it is seen that Broyden's method is fastest.

For the RFR not only single periodic states are computed, but also a branch of periodic states with varying cycle lengths is calculated. First the turning points were located by trial and error. Then a fixed list of cycle lengths is constructed. The cycle lengths in this list first increase up to the first turning point, then decrease to the second turning point and then increase again. At each of these cycle lengths in the list the periodic state is computed using both Broyden's method and the Newton-Picard method. The initial condition for each new computation at a new cycle length is calculated using a secant predictor: by linear extrapolation using the previous two computed periodic states the new initial condition is determined. Note that the new cycle length is not compute by linear extrapolation, but taken from the constructed list.

The branch of periodic states is shown in Fig. 4.4. Broyden's method needs 1121 cycle simulations to compute all the 76 solutions that construct the branch and the Newton-Picard method needs 3225 cycle simulations (these numbers are based on the residual convergence criterion). Thus the Newton-Picard method is almost three times as slow as Broyden's method. The Newton-Picard method, however, also provides information about the stability of the periodic states. In the subspace iteration approximations to the dominant Floquet multipliers are readily obtained without extra cycle simulations. In Fig. 4.5 the largest Floquet multiplier is plotted against the cycle length.

The above explained continuation method is a very simple method. There exist more sophisticated methods, such as the pseudo arc-length method, which we will use in Chapter 5. The pseudo arc-length continuation varies the step size along the branch of periodic solutions, depending on the convergence rate and domain of attraction of the method used for the computation of the periodic solutions. We used this simple method because we wanted to be sure that both Broyden's method and the Newton-Picard method computed exactly the same points on the branch. In this way there is no influence of the different domains of attraction for the different methods on the results of the comparison. In Chapter 5, where we use the pseudo arc-length method, we do study the influence of the domain of attraction on the computation of a branch of periodic solutions.

4.5 Do System Characteristics Determine the Method?

The systems discussed in the previous section are nonlinear systems and we have seen that it is difficult to determine from the characteristics of the system which method is most appropriate to compute the periodic state. If the system under consideration is linear, then a great deal more can be said about the convergence of this system to a periodic state and about the performance of the various methods to compute the periodic state.

To illustrate this fact we shall study two linear PSA systems in this section. The first system, System I, is a pressure swing adsorber used to remove water from air (see also Chapter 1), using alumina as the sorbent material. The adsorption is assumed to be 100% selective towards H_2O. The most important modeling assumptions are: the system is isothermal, the adsorption rate is described by the linear driving force model, H_2O is a trace component in an excess amount of air and the adsorption isotherm is linear. These four assumptions remove all the nonlinear terms from the model.

The second system, System II, is a pressure swing adsorber used to remove CO_2 from He using silica gel. The modeling assumptions for this system are exactly the same as for System I. It is seen from Table A.6 (see Section A.4) that the adsorption equilibrium constant for System II is much smaller than for System I. This means that the buffer capacity for System I is larger than for the System II. In addition, the adsorption rate constant for System I is two orders smaller than for System II. It is thus expected that a dynamic simulation of System II will approach a periodic state much faster than for System I.

4.5.1 Models for the H_2O/Air and CO_2/He PSA Systems

In this section the model equations and parameter values for the H_2O/air and CO_2/He PSA systems are given. The dimensionless model is the same for both systems. The variables in the model are the gas phase concentration, denoted by $c^{(i)} : [0,\infty) \times [0,1] \to [0,\infty)$, $(t,z) \to c^{(i)}(t,z)$ ($i = 1$: oxygen, $i = 2$: nitrogen), and the adsorbed amount, denoted by $q^{(i)} : [0,\infty) \times [0,1] \to [0,\infty)$, $(t,z) \to q^{(i)}(t,z)$. The mass balance and the adsorption rate equation are given by

$$\begin{aligned} c_t &= -K_1(vc)_z + K_2K_3(c-q)/d\ , \\ q_t &= K_3(c-q)/d\ . \end{aligned}$$

The relation between the three dimensionless parameters K_1, K_2 and K_3 and the physical parameters is given by

$$\begin{aligned} K_1 &= u_{\text{ads}}T/L, \\ K_2 &= \frac{1-\epsilon}{\epsilon}, \\ K_3 &= Tk_{\text{ads}}, \end{aligned}$$

Table 4.9: The periodic values of the velocity $v(t,z)$ and the pressure $d(t)$ and the boundary conditions for $c(t,z)$ for the H_2O/air and CO_2/He PSA systems

	$v(t,z)$	$d(t)$	$c(t,0)$	$c(t,1)$
pres. step	$\frac{d_t(t)}{K_1 d(t)}(z-1)$	$\frac{P_L}{P_H} + \frac{Tt}{t_{pre}}\left(1-\frac{P_L}{P_H}\right)$	$d(t)$	-
ads. step	1	1	1	-
blow. step	$\frac{d_t(t)}{K_1 d(t)}(z-1)$	$1-\left(1-\frac{P_L}{P_H}\right)\frac{Tt}{t_{blo}}$	-	-
purge step	-2	$\frac{P_L}{P_H}$	-	0

where v_{ads} is the gas velocity during the adsorption step used as the characteristic velocity, T the duration of one cycle, L the bed length, ϵ the bed porosity, K the adsorption equilibrium constant and k_{ads} the linear driving force rate constant during the adsorption step.

The parameters that change periodically are $d(t)$, $v(t,z)$ and the boundary concentrations of the adsorbate in the feed and purge streams (Table 4.9). The values of the physical parameters for the H_2O/air and CO_2/He PSA systems are given in Table A.6 in Section A.4.

4.5.2 Results

For the numerical simulation of one cycle of the H_2O/air or CO_2/He PSA system, first the spatial variable z is discretized on a equidistant grid of 256 nodes using first order upwind finite differences. The resulting system of 512 (256 nodes for the gas phase and 256 nodes for the solid phase) ordinary differential equations is integrated in time using a four stage Runge-Kutta method with 60000 steps for one cycle. Hence $N = 512$.

In Fig. 4.6 the residual versus the CPU time in seconds is given for the two linear PSA systems and in Table 4.5 the number of cycle simulations needed is listed. We use the convergence criterion based on the the residual ε_x with $\delta_0 = 10^{-12}$ and the convergence criterion based on the deviation γ_x defined by (4.3)-(4.4) with $\delta_0^* = 10^{-8}$. We see that System I converges very slowly to a periodic state, whereas System II converges very fast.

If the model equations for a system are linear then it follows that the period map $F : \mathbb{R}^N \to \mathbb{R}^N$ is affine, i.e., can be written as

$$F(x) = Ax + b\,, \tag{4.14}$$

where A is an $N \times N$ matrix and $b \in \mathbb{R}^N$. Note that a periodic state corresponds to a fixed point of F, i.e., a vector x that satisfies the equation

$$x = Ax + b. \tag{4.15}$$

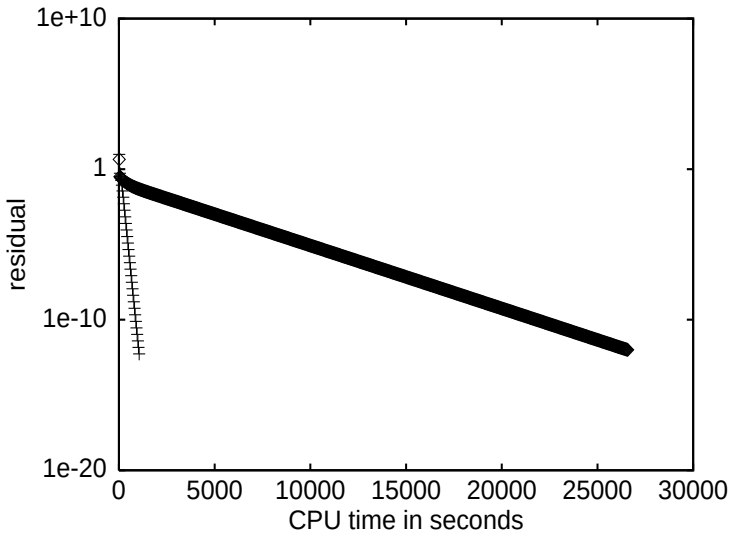

Figure 4.6: The residual versus the CPU time in seconds for the dynamic simulation for the H_2O/air PSA separation (*thick black line*, 769 iterations) and the CO_2/He PSA separation (+, 31 iterations).

This equation can also be written as $(A - I)x = -b$, where I is the identity matrix. If the matrix $(A - I)$ is invertible, then the equation (4.15) has a unique solution, given by $-(A-I)^{-1}b$, and the system has a unique periodic state. So if a stable periodic state of a linear system is found (stability means $|\lambda| < 1$ for all eigenvalues λ of A), it follows that $(A - I)$ is invertible and thus the periodic state must be unique.

From equation (4.15), it is seen that the dynamic simulation converges to a fixed point with convergence rate equal to the modulus of eigenvalue with largest modulus. of A. This means that if the eigenvalue with largest modulus of A is known, then it can be computed how many iterations the dynamic simulation needs in order to converge to a periodic state.

From Fig. 4.6, it is seen that the convergence rate of the dynamic simulation is indeed constant. For System I, the convergence rate equals 0.969 and for System II it equals 0.457. The eigenvalue with largest modulus of the Jacobian has modulus 0.9692739 for System I and 0.4575510 for System II.

For linear systems, it is known that Newton's method converges in one iteration. So, if N is known, it can also be computed how many cycle simulations Newton's method, with a finite difference approximation of the Jacobian, needs to determine the periodic state of a given system. Therefore we can conclude that for a *linear* system, when the modulus of the largest eigenvalue of the Jacobian and the dimension N is known, it is possible to determine on beforehand whether Newton's method or the dynamic simulation will be more efficient to compute the periodic state.

Because Newton's method converges theoretically in one iteration for linear systems and because the Newton-Picard method as implemented starts with one Newton iteration, the Newton-Picard method as implemented here reduces to Newton's method for $\rho < 1$. We also know that for $\rho = 1$ the Newton-Picard method reduces to dynamic simulation.

Therefore the discussion of the Newton-Picard method for linear systems is included in the discussion of Newton's method and dynamic simulation.

It is expected that Newton's method converges in one iteration to a periodic state, but due to the errors made in the finite difference approximation of the Jacobian, Newton's method needs a few more iterations (see Table 4.5), which are computationally efficient, since the same Jacobian can be used for each iteration. By comparing the results of the dynamic simulation with those for Newton's method, we conclude that Newton's method accelerates the convergence to a fixed point for System I, but not for System II.

It is seen that for both system Broyden's method converges much faster than either dynamic simulation or Newton's method. For Broyden's method, it is known that, for linear systems, the theoretical bound for convergence is $2N$ iterations [13]. For System I and System II this clearly is an overly conservative bound and one can ask whether better bounds exist given the special structure of the matrix A in (4.14). The disadvantage of Broyden's method when compared to Newton's method or dynamic simulation is that also for linear systems Broyden's method does not give any information about the stability of a computed periodic state.

4.6 Notes

The contents of this chapter are based on parts of the paper [61], and also on [57] and [62]

The convergence criterion based on the residual introduced in Section 4.2 is a commonly used criterion both in chemical engineering and in numerical analysis, see for example eq. (6) in [56], but also [50], [7], [15], [18] and [51].

The parameters and the model for the non-isothermal CO_2/N_2 PSA system discussed in Section 4.4.1, are taken from [29].

The H_2S/natural gas PSA system discussed in Section 4.4.2, as also been studied in [23] under a slightly different operating scheme. The physical and chemical parameters are all taken from the same paper [23].

Robustness problems for the Newton-Picard method, as described in the case of the H_2S/natural gas PSA system, can be solved not only by varying the value of ρ. Also the number of Picard iteration in **Step 7** of the Newton-Picard algorithm and the number of subspace iterations in **Step 6** influence the robustness of the Newton-Picard method. For an extensive study of the influence of all these parameters on the performance of the Newton-Picard algorithm, see [35, 36]. The Newton-Picard method can in fact be implemented in such a way that it behaves like Newton's method.

The model and parameters for the reverse flow reactor discussed in Section 4.4.3 are all taken from [12]

The model and parameters for the two linear PSA systems, discussed in Section 4.5 are taken from [30]. These systems are also discussed in combination with convergence acceleration methods in [31]

Chapter 5

A Broyden Rank $p+1$ Update Continuation Method with Subspace Iteration

5.1 Introduction

In this chapter we introduce a continuation method for the computation of branches of periodic solutions of periodically forced systems. The method combines the efficiency of Broyden's method with the ability of the subspace method to determine the stability of the computed periodic solutions.

A branch of periodic solutions is the curve in $\mathbb{R}^{N+1}$ defined by the equation

$$x = F(x, \lambda), \tag{5.1}$$

where the map $F : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$ is the period map depending on the scalar parameter λ. We define the Jacobian of F at the point $\boldsymbol{x} = (x, \lambda)$ to be the square matrix $J(x, \lambda)$ given by $J(x, \lambda) := \frac{\partial F}{\partial x}(x, \lambda)$. The stability of a solution x^* of (5.1) is determined by the eigenvalues of the Jacobian $J(x^*, \lambda)$ at the fixed point x^*.

We know that solutions $\boldsymbol{x} = (x, \lambda)$ of the equation $F(x, \lambda) = x$, form in general, at least locally, a curve in $\mathbb{R}^{N+1}$. Pseudo arc-length continuation allows to compute this curve also past turning points. Suppose we have a solution $\boldsymbol{x}_0 = (x_0, \lambda_0)$ of $F(x, \lambda) = x$, as well as the normalized direction vector of the solution curve at (x_0, λ_0), which we will denote by $\dot{\boldsymbol{x}}_0 = (\dot{x}_0, \dot{\lambda}_0)$. Pseudo arc-length continuation consists of solving the following equations for x_1 and λ_1

$$\begin{cases} F(x_1, \lambda_1) - x_1 = 0, \\ (x_1 - x_0)^T \dot{x}_0 + (\lambda_1 - \lambda_0)\dot{\lambda}_0 - s = 0, \end{cases} \tag{5.2}$$

where s is the step size and $(x_1 - x_0)^T$ is the transpose of $(x_1 - x_0)$. It is not always possible to compute the direction vector $\dot{\boldsymbol{x}}_0 = (\dot{x}_0, \dot{\lambda}_0)$ directly, but a good approximation

can be obtained from

$$\dot{\boldsymbol{x}}_0 \approx \frac{(\boldsymbol{x}_0 - \boldsymbol{x}_{-1})}{|(\boldsymbol{x}_0 - \boldsymbol{x}_{-1})|}, \tag{5.3}$$

where $\boldsymbol{x}_{-1}$ is a previously computed solution on the curve.

In this chapter we discuss a Broyden rank $p+1$ update continuation method with Subspace Iteration (in short the BSI rank $p+1$ method) to solve equations of the type (5.2). The method combines the efficiency of Broyden's method with the ability of the subspace iteration method to determine the stability of the computed solutions. The application of both the BSI rank $p+1$ method and the Newton-Picard method to an example system, shows that the BSI rank $p+1$ method is approximately twice as efficient as the Newton-Picard method, in terms of number of evaluations of F. The difference in efficiency lies mainly in the fact that the BSI method uses less evaluations of F per iteration than the Newton-Picard method. The example shows that the convergence rate and robustness of the two methods are comparable.

The organization of this chapter is as follows. In Section 5.2, we introduce the BSI rank $p+1$ method and discuss the merits of the method in comparison with the Newton-Picard method. In Section 5.3 we present the model equations for the cooled reverse flow reactor. In Section 5.4 we compute a branch of periodic states of the reverse flow reactor using both the BSI rank $p+1$ method and the Newton-Picard method, and compare the efficiency of both methods.

5.2 BSI Rank $p+1$ Method

In the Introduction we saw that, for the computation of a branch of periodic solutions of a periodically forced system, we have to solve repetitively systems of equations of the form

$$\begin{pmatrix} F(x,\lambda) - x \\ w^T(x-y) + \kappa(\lambda - \mu) - s \end{pmatrix} = 0, \tag{5.4}$$

where $w,\ y \in \mathbb{R}^N$ and $\kappa,\ \mu,\ s \in \mathbb{R}$ are given. For convenience, we define the map $G : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$ by

$$G(\boldsymbol{x}) = G(x,\lambda) := \begin{pmatrix} F(x,\lambda) - x \\ w^T(x-y) + \kappa(\lambda - \mu) - s \end{pmatrix}, \tag{5.5}$$

where $\boldsymbol{x}$ is the $N+1$ dimensional vector (x,λ).

An efficient way to compute solutions to the system $G(\boldsymbol{x}) = 0$ is to use Broyden's method (see Section 1.5.3). Broyden's method produces approximations to a zero of G using the following iteration scheme

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + H_i G(\boldsymbol{x}_i)\ , \tag{5.6}$$

with $H_i \in \mathbb{R}^{(N+1)\times(N+1)}$ iterative approximations to $-\frac{\partial G}{\partial \boldsymbol{x}}(\boldsymbol{x}_i)^{-1}$ defined by

$$H_{i+1} = H_i - \frac{(\boldsymbol{p}_i + H_i\boldsymbol{g}_i)\boldsymbol{p}_i^T H_i}{\boldsymbol{p}_i^T H_i \boldsymbol{g}_i}\,, \tag{5.7}$$

where $\boldsymbol{g}_i = G(\boldsymbol{x}_{i+1}) - G(\boldsymbol{x}_i)$ and $\boldsymbol{p}_i = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i$. This iteration scheme for H_i is derived from the following constraints on the update H_{i+1} of H_i:

$$-H_{i+1}^{-1}\boldsymbol{p}_i = \boldsymbol{g}_i, \tag{5.8}$$

$$-H_{i+1}^{-1}\boldsymbol{y} = -H_i^{-1}\boldsymbol{y} \quad \text{for all } \boldsymbol{y} \perp \boldsymbol{p}_i. \tag{5.9}$$

Equation (5.8) can be viewed as a secant approximation of $\frac{\partial G}{\partial \boldsymbol{x}}(\boldsymbol{x}_{i+1})$ in the direction $\boldsymbol{p}_i$. Note that the only information that is used in updating the approximation H_i at the new approximation $\boldsymbol{x}_{i+1}$ is the function value $G(\boldsymbol{x}_{i+1})$. The method thus uses only one evaluation of F in each iteration. However, there is no guarantee that, if the approximations $\boldsymbol{x}_i$ converge to a solution $\boldsymbol{x}^*$ of $G(\boldsymbol{x}) = 0$, the approximations H_i converge to $-\frac{\partial G}{\partial \boldsymbol{x}}(\boldsymbol{x}^*)^{-1}$. This means that the approximations H_i cannot be used to obtain good approximations of the eigenvalues of $\frac{\partial F}{\partial x}(\boldsymbol{x}^*)$, which are needed for the determination of the stability of the solution $\boldsymbol{x}^*$. In this section we discuss an approach to combine Broyden's method with a subspace iteration algorithm, so that also approximations to the largest eigenvalues of $\frac{\partial F}{\partial x}(\boldsymbol{x}^*)$ can be computed accurately.

For this approach we do not only need the initial approximations $\boldsymbol{x}_0$ and H_0, required for Broyden's method, but also initial approximations to the p largest eigenvalues of $\frac{\partial F}{\partial x}(\boldsymbol{x}_0)$ together with an initial approximation $V_0 \in \mathbb{R}^{N\times p}$ for the orthonormal basis of the subspace spanned by the p corresponding eigenvectors. In the first step of our new approach, we proceed as in Broyden's method, and compute the next approximation to the solution of (5.4) with the same formula as before

$$\boldsymbol{x}_1 \;=\; \boldsymbol{x}_0 + H_0 G(\boldsymbol{x}_0). \tag{5.10}$$

Now we would like to update H_0, V_0 and the approximations of the eigenvalues.

Let us first consider the updating of the eigenvalues and of V_0. We would like the updates of the eigenvalues to approximate the p largest eigenvalues of the Jacobian $\frac{\partial F}{\partial x}(\boldsymbol{x}_1)$ in the new approximation $\boldsymbol{x}_1$ and the update V_1 to approximate the basis of the subspace spanned by the corresponding eigenvectors. Both the updates of the eigenvalues and of V_0 can be obtained by performing one (or more) subspace iteration(s). We use the same version of the subspace iteration algorithm as presented in Chapter 3. In each iteration of the BSI rank $p+1$ method, we only perform one iteration of the subspace iteration algorithm. This iteration provides us with new approximations to the largest eigenvalues of $\frac{\partial F}{\partial x}(\boldsymbol{x}_1)$, with a new orthonormal basis $V_1 = \tilde{V}_1$ and with the matrix

$$W_0 = \frac{\partial F}{\partial x}(\boldsymbol{x}_1)\tilde{V}_0 = \frac{\partial F}{\partial x}(\boldsymbol{x}_1)V_0. \tag{5.11}$$

This last matrix is very useful in updating H_0. If we define the matrices

$$\boldsymbol{V} := \begin{pmatrix} V_0 \\ 0 \end{pmatrix} \text{ and } \boldsymbol{Z} := \begin{pmatrix} W_0 - V_0 \\ w^T V_0 \end{pmatrix}, \tag{5.12}$$

then it is easily seen that

$$\frac{\partial G}{\partial \boldsymbol{x}}(\boldsymbol{x}_1)\boldsymbol{V} = \boldsymbol{Z}. \tag{5.13}$$

We can use the equation (5.13) in the same way as the constraint (5.8) is used in Broyden's method, to update H_0. So, analogously to the Broyden rank one update, we would now like to update H_0 in such a way that

$$-H_1^{-1}\boldsymbol{V} = \boldsymbol{Z}, \tag{5.14}$$
$$-H_1^{-1}(\boldsymbol{x}_1 - \boldsymbol{x}_0) = \big(G(\boldsymbol{x}_1) - G(\boldsymbol{x}_0)\big), \tag{5.15}$$
$$H_1^{-1}\boldsymbol{q} = H_0^{-1}\boldsymbol{q} \text{ for all } \boldsymbol{q} \text{ with } \boldsymbol{V}^T\boldsymbol{q} = 0 \text{ and } (\boldsymbol{x}_1 - \boldsymbol{x}_0)^T\boldsymbol{q} = 0. \tag{5.16}$$

If $(I - \boldsymbol{V}\boldsymbol{V}^T)(\boldsymbol{x}_1 - \boldsymbol{x}_0) \neq 0$, then the p columns of $\boldsymbol{V}$ and the vector $(\boldsymbol{x}_1 - \boldsymbol{x}_0)$ are linearly independent and span a $p+1$ dimensional subspace of $\mathbb{R}^{N+1}$, which we denote by $\boldsymbol{U}$. Let the $(N+1) \times (N+1)$ matrix $\boldsymbol{Y}$ be such that its first p columns are equal to minus the columns of $\boldsymbol{V}$, its $(p+1)$th column is equal to the vector $-(\boldsymbol{x}_1 - \boldsymbol{x}_0)$ and its remaining $N-p$ columns are a basis of the orthogonal complement of $\boldsymbol{U}$. Then the matrix $\boldsymbol{Y}$ has by definition rank $N+1$ and is invertible. We define the $(N+1) \times (N+1)$ matrix $\boldsymbol{X}$ to be the matrix with the first p columns equal to the columns of $\boldsymbol{Z}$, the $(p+1)$th columns equal to $\big(G(\boldsymbol{x}_1) - G(\boldsymbol{x}_0)\big)$ and the remaining columns equal to the last $N-p$ columns of $H_0^{-1}\boldsymbol{Y}$. If H_0 is invertible this last matrix is well defined. Now we can write the conditions (5.14)-(5.16) for H_1^{-1} as

$$H_1^{-1}\boldsymbol{Y} = \boldsymbol{X},$$

where $\boldsymbol{Y}$ is invertible. Thus, if H_0 is invertible and $(I - \boldsymbol{V}\boldsymbol{V}^T)(\boldsymbol{x}_1 - \boldsymbol{x}_0) \neq 0$, the matrix H_1^{-1} characterized by the equations (5.14)-(5.16) is well defined and unique.

If H_1^{-1}, defined by (5.14)-(5.16), is invertible, then its inverse H_1 can be computed as follows. First we define $B_0 := H_0$ and then recursively

$$B_{l+1} \;:=\; B_l - \frac{(\boldsymbol{v}_l + B_l\boldsymbol{z}_l)\boldsymbol{v}_l^T B_l}{\boldsymbol{v}_l^T B_l \boldsymbol{z}_l}, \tag{5.17}$$

up to B_p, where $\boldsymbol{v}_l$ and $\boldsymbol{z}_l$ denote the lth column of respectively $\boldsymbol{V}$ and $\boldsymbol{Z}$. Now, the update H_1 is given by

$$H_1 \;=\; B_p - \frac{(\boldsymbol{p} + B_p\boldsymbol{f})\boldsymbol{p}^T B_p}{\boldsymbol{p}^T B_p \boldsymbol{f}}, \tag{5.18}$$

where we use the notation

$$\boldsymbol{p} \;=\; (I - \boldsymbol{V}\boldsymbol{V}^T)(\boldsymbol{x}_1 - \boldsymbol{x}_0), \tag{5.19}$$
$$\boldsymbol{f} \;=\; G(\boldsymbol{x}_1) - G(\boldsymbol{x}_0) - \boldsymbol{Z}\boldsymbol{V}^T(\boldsymbol{x}_1 - \boldsymbol{x}_0). \tag{5.20}$$

At this point we have computed the new updates $\boldsymbol{x}_1$, V_1, H_1 and updates of the p largest eigenvalues. By following the above procedure starting with these updates, we

can again obtain another set of updates, and so on. This iteration of the above described procedure defines the BSI rank $p+1$ method, and can be summarized by the following steps:

BSI rank $p+1$ method

Step 1: *Supply $H_0 \in \mathbb{R}^{(N+1)\times(N+1)}$, invertible; $V_0 \in \mathbb{R}^{N\times p}$, orthonormal; $\boldsymbol{x}_0 = (x_0, \lambda_0) \in \mathbb{R}^{(N+1)}$, and set $i = 0$.*

Step 2: *Compute*

$$\boldsymbol{x}_{i+1} \;=\; \boldsymbol{x}_i + H_i G(\boldsymbol{x}_i).$$

(costs one evaluation of F)

Step 3: *Subspace iteration: compute $S := \frac{\partial F}{\partial x}(\boldsymbol{x}_{i+1})V_i$ (costs p evaluations of F). Compute the ordered Schur-factorization $Y^T R Y$ of $V_i S$. Compute $V = SY$. Orthonormalize the columns of V and put the result in V_{i+1}.*

Step 4: *Compute*

$$H_{i+1} \;=\; \begin{cases} B_p - \dfrac{(\boldsymbol{p} + B_p\boldsymbol{f})\boldsymbol{p}^T B_p}{\boldsymbol{p}^T B_p \boldsymbol{f}} & \text{if } \boldsymbol{p}^T B_p \boldsymbol{f} \neq 0 \\ & \text{and } (I - \boldsymbol{Z}\boldsymbol{Z}^T)\big(G(\boldsymbol{x}_{i+1}) - G(\boldsymbol{x}_i)\big) \neq 0, \\ B_p & \text{otherwise,} \end{cases}$$

where B_p is defined recursively by $B_0 = H_i$ and

$$B_{l+1} = \begin{cases} B_l - \dfrac{(\boldsymbol{v}_l + B_l\boldsymbol{z}_l)\boldsymbol{v}_l^T B_l}{\boldsymbol{v}_l^T B_l \boldsymbol{z}_l} & \text{if } \boldsymbol{v}_l^T B_l \boldsymbol{z}_l \neq 0, \\ B_l & \text{if } \boldsymbol{v}_l^T B_l \boldsymbol{z}_l = 0, \end{cases}$$

with $\boldsymbol{v}_l$ and $\boldsymbol{z}_l$ the lth column of respectively

$$\boldsymbol{V} := \begin{pmatrix} V_i \\ 0 \end{pmatrix} \text{ and } \boldsymbol{Z} := \begin{pmatrix} \big(\frac{\partial F}{\partial x}(\boldsymbol{x}_{i+1}) - I\big)V_i \\ w^T V_i \end{pmatrix}.$$

and where

$$\begin{aligned} \boldsymbol{p} &= (I - \boldsymbol{V}\boldsymbol{V}^T)(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i), \\ \boldsymbol{f} &= G(\boldsymbol{x}_{i+1}) - G(\boldsymbol{x}_i) - \boldsymbol{Z}\boldsymbol{V}^T(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i). \end{aligned}$$

Set $i = i + 1$, go to **Step 2**..

This algorithm needs $p+1$ evaluations of F per iteration. In Section 5.4 we will use the algorithm to compute a branch of periodic states of a reverse flow reactor.

When we compute the first periodic state on the branch, we supply in **Step 1** of the algorithm $H_0 = I$. With this choice the first iteration is of the BSI rank $p+1$ method is equal to a Picard iteration. For the computation of the following periodic states on the branch, we take H_0 to be equal to the H_i obtained at the end of the computation of the previous periodic state.

In order to give an idea of the efficiency of the BSI rank $p+1$ method, we compare its performance with the performance of two variants of the Newton-Picard method that are suited for pseudo-arclength continuation, These two variants are introduced in [37]. The first variant of the method is called CNP in [37]. This variant uses the the Sherman-Morrison formula in combination with the Newton-Picard method to solve (5.4). This variant needs $p+2l+4$ evaluations of F per iteration, where l is a parameter of the method that specifies the number of Picard iterations used in one full iteration of the Newton-Picard method. The second variant is called CNPGS. This variant sets some non-zero elements of the Jacobian of G equal to zero, such that the Newton-Picard method can directly be applied to the resulting linear equations. This variant needs $p+l+2$ evaluations of F per iteration and so is less expensive than CNP. In [37] it is reported that the CNP and CNPGS have the same asymptotic convergence rate, but that CNP is the more robust method (i.e., has a larger domain of attraction) and thus allows larger step sizes in the continuation algorithm.

In [37], the subspace iteration is performed on $p+p_e$ (with $p_e = 2$, 3 or 4) vectors in order to accelerate the convergence of the subspace iteration procedure. In [37], also the number p is varied during the computations in order to ensure that the subspace iteration is only performed on the subspace spanned by eigenvectors of which the corresponding eigenvalues are outside the region $D_\rho := \{z \in \mathbb{C} : |z| < \rho\}$ (usually ρ is taken to be 0.5). Here, for both the Newton-Picard methods and the BSI method, we only perform the subspace iteration on p vectors, and keep this number fixed during the computations.

In [37] it is reported that the Newton-Picard method converges linearly. Broyden's method has theoretically better convergence properties. In [4] it is shown that Broyden's method converges locally at least Q-superlinearly, and in [13] it is shown that the method enjoys local $2N$-step, Q-quadratic convergence for nonlinear problems. We do not know whether these results also hold for the BSI rank $p+1$ method.

An important issue for a numerical method is the memory usage. The BSI rank $p+1$ method as discussed above needs to store the $N \times N$ matrix H_i. For the example processes presented in this thesis, the N is not so large that the storage of H_i becomes a problem. One can think, however, of model equations of processes for which the discretization produces an N so large that it is impossible to store an $N \times N$ matrix. This is, for example, the case when in model equations the radial dimension of the reactor or adsorber is incorporated. As an alternative to storing the whole H_i, one can start with $H_0 = I$, which does not require much memory, and then storing each rank $p+1$ update in the BSI rank $p+1$ method using $2(p+1)$ vectors. This approach, however, still leads to memory

problems when the BSI rank $p+1$ method needs many iterations to converge.

To overcome this problem for Broyden's method there exist variants of the method with limited memory use [39]. The idea is to approximate H_i, when the number of updates becomes too large, with a lower rank matrix, so that the required storage remains limited. We believe that such an approach is also possible for the BSI rank $p+1$ method, although we do not pursue this matter further here.

The largest matrix that Newton-Picard method requires to be stored, is the $N \times p$ matrix V_i that is an approximation to a basis of the subspace spanned by the eigenvectors corresponding to the p largest eigenvalues. Here the p is essentially independent of the discretization, so that the storage used by the Newton-Picard method depends only linearly on N.

5.3 The Cooled Reverse Flow Reactor

We use the BSI rank $p+1$ method and the CNP and CNPGS methods to compute a branch of periodic states of a cooled reverse flow reactor. In the RFR occurs a single irreversible, exothermic, first order reaction. We describe the RFR by a one-dimensional, pseudo-homogeneous model that accounts for axial heat and mass dispersion and external mass transfer resistance between the fluid and the catalyst. The basic model assumption is that all the physical properties are independent of the temperature and the concentration. We use the same model equations as presented in Chapter 3: the equations (3.14)-(3.15) with the boundary conditions (3.16) describe the model. Here the parameter K_4, that represents the cooling capacity, is used as the bifurcation parameter.

A periodic state of (3.14)-(3.15) with the boundary conditions (3.16) is a solution that satisfies

$$\theta(0,z) = \theta(1,1-z) \quad \text{and} \quad \chi(0,z) = \chi(1,1-z). \tag{5.21}$$

The values of the dimensionless parameters are given in Table 3.2. The values of the physical parameters are given in Table A.2, and the relations between the physical parameters and the dimensionless parameters are given in Section A.2.

5.4 Results and Comparison

For the reverse flow reactor we chose to discretize the spatial variable on a grid of 60 nodes using a finite volume approach with upwinding of the first space derivatives in the mass balance. The discretization of the model equations results for the RFR in a system of 120 ordinary differential equations. Hence $N = 120$. The system of ODE's is integrated in time using the NAG fortran library routine D02EJF. In this section we present the results for the different methods used to compute a branch of periodic states of the reverse flow reactor. The branch is depicted in Figure 5.4. Here stable periodic states are represented

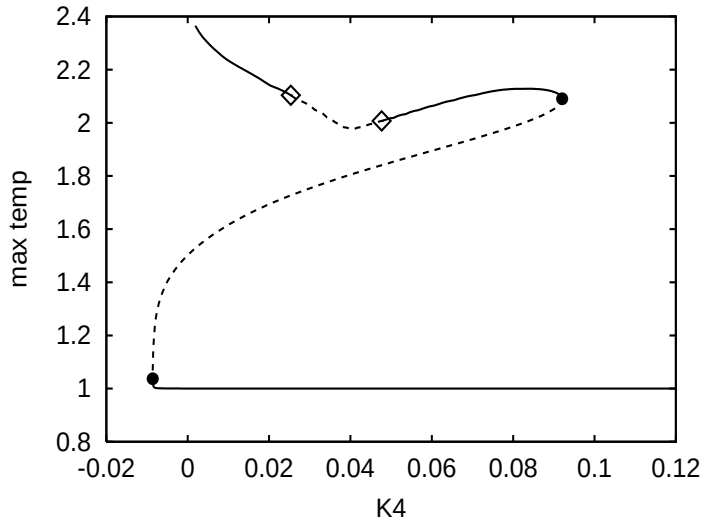

Figure 5.1: Branch of periodic solutions for the reverse flow reactor. Along the x-axis the parameter K_4 is plotted and along the y-axis the maximal dimensionless temperature in the bed at flow reversal is plotted. solid line: stable periodic states, dashed line: unstable states. $\diamond$: an eigenvalue crosses the unit circle at -1, $\bullet$: an eigenvalue crosses the unit circle at 1.

by solid lines and unstable states by a dashed line. The different bifurcation points are denoted in the figure and explained in the caption.

We used three different values of p, i.e. 5, 7 and 9, for all the methods. For the CNP and the CNPGS we also varied the parameter l.

For all methods we used the same simple step size control procedure. After a convergence failure, the step size is decreased and divided by two and when a point is computed sufficiently fast, the step size is increased and multiplied by 1.6 (up to a maximum step size). This variable step size strategy allows us to observe not only differences in convergence speed, but also differences in the size of the domain of attraction of the different methods. For each point (x, λ) computed on the branch, the three largest eigenvalues of $J(x, \lambda)$ (according to absolute value) are computed with four digits of accuracy. For this purpose after the BSI or Newton-Picard iterations in most cases only one subspace iteration was sufficient.

The number of evaluations of F that each method needs to compute the whole branch is listed in Table 5.1. The number of points computed on the curve are listed in Table 5.2. We clearly see that the BSI rank $p+1$ method is the most efficient method. Actually, the BSI rank $p+1$ method is most efficient for $p = 5$, but also for the other values of p the method is much more efficient than the either the CNP or the CNPGS method.

The next most efficient method is CNP with $l = 2$ and $p = 7$. Note that there is not much difference between the $p = 7$ and $p = 9$ cases. For $p = 5$, however, the CNP method needs considerably more function evaluations.

The CNPGS method is the least efficient method and needs the most function evaluations. This is clearly due to the fact that it has the smallest domain of attraction. As

Table 5.1: Number of evaluations of F for the different methods

method	# F eval. for $p = 5$	# F eval. for $p = 7$	# F eval. for $p = 9$
BSI	11551	12228	13557
CNP (l=1)	40296	22268	20900
CNP (l=2)	32551	19992	20744
CNP (l=3)	26504	20832	21964
CNPGS (l=1)	30784	28428	29032
CNPGS (l=2)	31292	30266	29623
CNPGS (l=3)	27797	28091	31550

Table 5.2: Number of continuation points for the different methods

method	# points for $p = 5$	# points for $p = 7$	# points for $p = 9$
BSI	342	341	340
CNP (l=1)	464	355	359
CNP (l=2)	420	345	359
CNP (l=3)	356	350	360
CNPGS (l=1)	415	411	396
CNPGS (l=2)	419	438	408
CNPGS (l=3)	403	395	409

is seen from Table 5.2, the CNPGS method needs considerably more continuation points than both the BSI and the CNP method. This fact is also illustrated in Figure 5.4. From this figure, we observe that the BSI rank $p + 1$ method uses the maximum step size all along the branch except close to the turning points. The CNP method is less robust, as the method uses smaller step sizes, than the BSI rank $p+1$ method in a neighbourhood of the left turning point. The CNPGS method is the least robust method, since it is the only method that uses smaller step sizes to compute the unstable part of the branch between the two points where an eigenvalues crosses the unit circle at -1.

In [37] it is noted that the performance of the Newton-Picard methods in terms of the number of evaluations of F needed, is essentially independent of the number of nodes used in the discretization. In the order to check whether this is also the case for the BSI rank $p + 1$ method, we computed the same branch of solutions using a discretization of model equations on 100 nodes, so that $N = 200$. In this case the BSI rank $p + 1$ method with $p = 7$ needed 12714 evaluations of F to compute the whole branch. This number is close to the number of evaluations of F needed for the discretization on 60 nodes.

5.5 Notes

The contents of this chapter are based on [58].

To see that the solutions of $x = F(x, \lambda)$ form a curve in $\mathbb{R}^{N+1}$, the implicit function theorem can be used, see for example [14].

A good discussion of pseudo-arclength continuation can be found in [28].

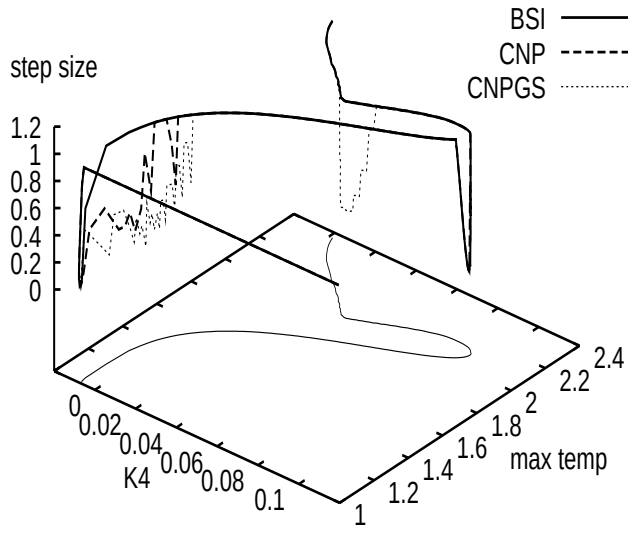

Figure 5.2: The step sizes for the different methods along the branch. Along the x-axis the parameter K_4 is plotted, along the y-axis the maximal dimensionless temperature in the bed at flow reversal is plotted and along the z-axis the step size used to compute the point on the branch is plotted. The maximum step size is equal to one. For each method the most efficient run is used; BSI: $p = 7$, CNP: $l = 2$ and $p = 7$, CNPGS: $l = 3$ and $p = 5$.

The approach of performing the subspace iteration on p vectors, and keeping this number fixed during the computations, works nicely for the cooled reverse flow reactor (see also Fig. 3.8). However, for dynamical systems with more wildly behaving eigenvalues, a procedure for varying p as used in [49, 20] or in [37] is necessary.

Another difference between our approach and that in [37] is that we do not use locking and deflation in the subspace iteration. Also the matrix-vector products $\frac{\partial F}{\partial x}(\boldsymbol{x}_{i+1})V_i$ in **Step 3** in the algorithm above, are calculated using a finite difference approximation given by

$$\frac{\partial F}{\partial x}(\boldsymbol{x})v \approx ||v||\epsilon^{-1}\big(F(x + \frac{\epsilon}{||v||}v, \lambda) - F(x, \lambda)\big),$$

instead of solving a variational form of the periodically forced system.

The model equations and the parameter values for the cooled reverse flow reactor are taken from [21].

Chapter 6

A Newton-Picard Optimal Control Algorithm

6.1 Introduction

This chapter is devoted to the numerical optimization of periodic adsorbers and reactors. We would like to make the following distinction between two different kinds of optimization: the optimization with respect to time-independent parameters and the optimization with respect to time-dependent parameters, also known as optimal control. In this chapter we would like to focus on the numerical optimal control of periodic processes, although we also optimize the presented example systems with respect to time-independent parameters. The reason for this is that for time-independent parameter optimization a large number of different methods and algorithms are readily available in many software packages and libraries, whereas the time-dependent parameter optimization, or optimal control, of periodic systems is much less a standard task.

The numerical optimization of a periodic process is typically aiming to minimize operating costs, or to maximize product yield, while maintaining certain product specifications, such as a minimum purity. In this chapter an efficient numerical optimization procedure for the computation of solutions to periodic optimal control problems is presented. The procedure is able to compute the optimal control of a periodic process under one or more nonlocal constraints, such as the minimum purity constraint. The presented method consists of a combination of the Newton-Picard shooting method with a first order gradient method.

We use the in this chapter presented numerical method to optimize a rapid pressure swing adsorber and a rapid pressure swing reactor. We also show that the results of a time-dependent parameter optimization can help to gain insight in the solution of a time-independent parameter optimization problem. For the computation of the solution to the time-independent parameter optimization problems we use a sequential quadratic programming method (NAG fortran library routine E04UCF)

The organization of this chapter is as follows. In Section 6.2, we present the numerical

method, which consists of the combination of a first order gradient method with the Newton-Picard method. In Section 6.2, we also discuss why this combination of methods is so efficient. In Section 6.3, we discuss the model equations of a rapid pressure swing adsorber and of a rapid pressure swing reactor and give the results of the optimization of these two systems.

6.2 Numerical Periodic Optimal Control

In this section we will start with an introduction to the mathematical formulation of the optimal control problem and we will formulate the first order gradient algorithm that we use to solve the optimal control problem.

Subsequently the efficient interaction between the first oder gradient method and the Newton-Picard method is discussed in detail. This concerns in particular the steps of the gradient method in which the Newton-Picard method is used to compute periodic solutions to the state and adjoint equations.

6.2.1 Problem Formulation

The common approach to model a pressure swing adsorber or reactor leads to a system of partial differential equations with periodic boundary conditions. We have seen that, in order to investigate the behaviour of the solutions to this system of partial differential equations numerically, the system has to be discretized in space. This leads to a large system of N (where N depends on the space discretization) time periodic ordinary differential equations. This system can be written as

$$\frac{dx}{dt}(t) = f(t, x(t), u(t)) \ , \quad \text{where } \ f(t+1, \cdot, u(t+1)) = f(t, \cdot, u(t)) \ , \tag{6.1}$$

where $x(t)$ is a N dimensional vector, $u(t)$ is a scalar 1-periodic control, and where $f(t, \cdot, u(t))$ is a function from the N dimensional vectors (denoted by $\mathbb{R}^N$) to the N dimensional vectors.

Since for optimization purposes we are only interested in periodic states of the system, we can restrict ourselves to consider solutions of (6.1) for $t \in [0, 1]$, supplemented with the periodic boundary conditions

$$x(0) = x(1). \tag{6.2}$$

The values that $u(t)$ can attain are restricted to a given interval B, say $B := [a, b]$, i.e.,

$$a \leq u(t) \leq b \text{ for all } t \in [0, 1]. \tag{6.3}$$

Furthermore, given the functionals L and M, depending on time, the state variables and the control parameter. we define the performance index

$$I(x, u) = \int_0^1 L(t, x(t), u(t))\, dt, \tag{6.4}$$

and a non-local constraint

$$J(x,u) = \int_0^1 M(t,x(t),u(t))\,dt = 0, \tag{6.5}$$

The optimal control problem, denoted by (OPC), now consists of finding a control $u(t)$ that minimize the performance index (6.4) under the constraints (6.1), (6.2), (6.3) and (6.5).

6.2.2 A First Order Gradient Method

In this section we describe the basic steps of the numerical procedure that we have implemented. We start with an overview of the procedure and then in the next section we concentrate on **Step 2** and **Step 3**, since in these two steps we use the Newton-Picard method as a new approach to make the first order gradient method more efficient.

Optimal Control Algorithm

Step 1: *Choose an initial control $u_0(t)$ and two positive scalars ϵ_1 and ϵ_2. Set $v(t) = u_0(t)$.*

Step 2: *Compute the periodic solution $x(t)$ of*

$$\frac{dx}{dt}(t) = f(t,x(t),v(t)), \quad x(0) = x(1).$$

Step 3 *Compute solutions $\Lambda(t)$ and $R(t)$ of*

$$\frac{d\Lambda^T}{dt}(t) = -\Lambda^T(t)\frac{\partial f}{\partial x}(t,x(t),v(t)) - \frac{\partial L}{\partial x}(t,x(t),v(t)), \tag{6.6}$$

$$\frac{dR^T}{dt}(t) = -R^T(t)\frac{\partial f}{\partial x}(t,x(t),v(t)) - \frac{\partial M}{\partial x}(t,x(t),v(t)), \tag{6.7}$$

with the periodic boundary conditions $\Lambda(0) = \Lambda(1)$ and $R(0) = R(1)$.

Step 4: *Compute the integrals*

$$I_1 = \int_\Omega \left[R^T\frac{\partial f}{\partial u} + \frac{\partial M}{\partial u}\right]\left[R^T\frac{\partial f}{\partial u} + \frac{\partial M}{\partial u}\right]^T dt,$$

$$I_2 = \int_\Omega \left[\Lambda^T\frac{\partial f}{\partial u} + \frac{\partial L}{\partial u}\right]\left[R^T\frac{\partial f}{\partial u} + \frac{\partial M}{\partial u}\right]^T dt,$$

$$I_3 = \int_\Omega \left[\Lambda^T\frac{\partial f}{\partial u} + \frac{\partial L}{\partial u}\right]\left[\Lambda^T\frac{\partial f}{\partial u} + \frac{\partial L}{\partial u}\right]^T dt,$$

where $\Omega \subseteq [0,1]$ with $\Omega = \{t \in [0,1] \mid a < u_0(t) < b\}$.

Step 5: *Compute* Δu

$$\Delta u(t) = -\epsilon_2 \Big[\frac{\partial L}{\partial u}(t, x(t), v(t)) + \xi \frac{\partial M}{\partial u}(t, x(t), v(t)) + [\Lambda(t) + R(t)\xi]^T \frac{\partial f}{\partial u}(t, x(t), v(t)) \Big],$$

with

$$\xi = (\epsilon_2 I_1)^{-1} (\epsilon_1 \int_0^1 M(t, x(t), v(t))\, dt - \epsilon_2 I_2).$$

Set $u(t) = v(t) + \Delta u(t)$ *and define the new control* $v(t) = u(t) + \max(a - u(t), 0) - \max(u(t) - b, 0)$. *This is done to make sure that the new control is in* $[a, b]$. *Go to* **Step 2**.

Note that the performance of the algorithm depends on the so-called damping factors ϵ_1 and ϵ_2. If the algorithm does not converge, then ϵ_1 and ϵ_2 should be chosen smaller. If the algorithm converges very slowly then ϵ_1 and ϵ_2 should be chosen larger, see [5]. For the two processes discussed in the sequel, we used the values $\epsilon_1 = 1$ and $\epsilon_2 \in [0.05, 0.5]$.

6.2.3 Computing Periodic Solutions of the State and Adjoint Equations

In **Step 2** and **Step 3** of the algorithm introduced in Section 6.2.2, we have to find a periodic solution $x(t)$ of the state equation (6.1) and periodic solutions $\Lambda(t)$ and $R(t)$ of the adjoint equations (6.6) and (6.7). In this section we discuss the Newton-Picard method that can be used to compute such periodic solutions.

The efficiency of the combination of the first order gradient method with the Newton-Picard method, lies in the fact that the Newton-Picard method does not only compute a periodic solution of the state equations (6.1), but also obtains information about the stability of this periodic solution. This information is used to compute very efficiently solutions to the adjoint equations (6.6) and (6.7).

First we discuss solutions of (6.1). We introduce, as before, the period map $F_u : \mathbb{R}^N \to \mathbb{R}^N$ that for a given control $u(t)$ assigns to the initial data at time zero, $x(0) = x_0$, the value of the solution of (6.1) after one cycle, i.e.,

$$F_u(x_0) = x(1, x_0),$$

where $x(t, x_0)$ is the solution of equation (6.1) with initial condition $x(0) = x_0$. The subscript u expresses the fact that the map F_u now also depends on the control $u(t)$. In order to determine a periodic solution of (6.1), we use the Newton-Picard method, as discussed in Chapter 3, to compute a fixed point of the map F_u.

Now we will look more closely at solutions of the adjoint equations. We will concentrate on the equation (6.6), equation (6.7) can be treated analogously. In the same manner as

for (6.1), we can define the period map associated to the the adjoint equation (6.6). This map, which we denote by $K : \mathbb{R}^N \to \mathbb{R}^N$, maps the initial condition $\Lambda(0) = \Lambda_0$ to $\Lambda(1)$. For the evaluation of this period map, we need to integrate the equation (6.6) with initial condition Λ_0. But note that this time integration is highly unstable in forward time for the chemical processes we have been studying. The reason is the minus sign in front of the terms $-\Lambda^T(t)\frac{\partial f}{\partial x}$ and $-R^T(t)\frac{\partial f}{\partial x}$ in equation (6.6): in the model equations for cyclic processes, there are usually diffusion terms, and these terms appear also in the adjoint equations, but here with a minus sign. This means that there exist solutions of the adjoint equations that grow very rapidly in forward time (see e.g. [5]).

One way to overcome this problem, is to integrate the equation (6.6) backward in time. In this way we do not evaluate the period map K, but the inverse of the period map, denoted by K^{-1}. For the computation of a periodic solution of the adjoint equation (6.6), we can now solve $(K^{-1}(\Lambda) - \Lambda) = 0$, instead of $(K(\Lambda) - \Lambda) = 0$.

Let us study the period map K and its inverse K^{-1} in more detail. Note that if $A(t)$ solves the initial value problem

$$\begin{aligned} &\frac{dA}{dt}(t) = \frac{\partial f}{\partial x}(t, x(t), u(t))A(t), \\ &A(0) = I. \end{aligned}$$

then $A(1) = J(x(0))$, where $J(x(0)) = \frac{\partial F_u}{\partial x}(x(0))$ denotes the Jacobian of F_u at $x(0)$. This allows us, using the variation of constants formula, and the identity

$$\frac{d}{dt}(A(t)^{-1}) = -A(t)^{-1}\frac{\partial f}{\partial x}(t, x(t), u(t))$$

to write the solution to the adjoint equation (6.6) with initial value $\Lambda(0) = \Lambda_0$ as follows

$$\Lambda(t)^T = \Lambda_0^T A(t)^{-1} - \int_0^t \frac{\partial L}{\partial x}(s, x(s), u(s))A(s)\, ds\ A(t)^{-1}. \tag{6.8}$$

Consequently, the period map K associated with equation (6.6) satisfies

$$\begin{aligned} K(\Lambda_0) = \Lambda(1) &= \left[\Lambda_0^T A(1)^{-1} - \int_0^1 \frac{\partial L}{\partial x}(s, x(s), u(s))A(s)\, ds\ A(1)^{-1}\right]^T \\ &= \left[\Lambda_0^T J(x(0))^{-1} - \int_0^1 \frac{\partial L}{\partial x}(s, x(s), u(s))A(s)\, ds\ J(x(0))^{-1}\right]^T. \end{aligned}$$

A simple computation shows that

$$K^{-1}(\Lambda(1)) = \Lambda_0 = \left[\Lambda(1)^T J(x(0)) + \int_0^1 \frac{\partial L}{\partial x}(s, x(s), u(s))A(s)\, ds\right]^T. \tag{6.9}$$

Thus, in order to find a periodic solution of (6.6), i.e., a solution that satisfies $\Lambda(0) = \Lambda(1)$, it suffices to solve

$$\Lambda^T\big(J(x(0)) - I\big) = g \tag{6.10}$$

for Λ, where

$$g = -\int_0^1 \frac{\partial L}{\partial x}(s, x(s), u(s))A(s)ds.$$

The important observation we can make from (6.9), is that the inverse period map for the adjoint equations is an affine map with Jacobian equal to the transposed Jacobian $J(x(0))^T$ of the period map F_u corresponding to the state equation at the periodic solution $x(t)$. In what follows, it is shown that this is a useful fact for the application of the Newton-Picard method to the adjoint equations.

Now suppose that the Newton-Picard method computed the periodic solution $x(t)$ of (6.1). Thus we have $F_u(x(0)) = x(0)$. In Chapter 3, we have seen that the Newton-Picard method is constructed in such a way that it gives approximations to the p largest eigenvalues of the Jacobian $J(x(0))$ of F_u at $x(0)$ and an orthonormal basis V_p of the subspace U that is spanned by the eigenvectors and generalized eigenvectors that correspond to these p largest eigenvalues. In addition we know the Jacobian $J(x(0))$ restricted to the subspace U, which can be written as

$$S := V_p^T J(x(0))V_p.$$

In order to solve the equation (6.10), we decompose Λ into a component in U and a component in the orthogonal complement of U, denoted by $U^\perp$. Let V_r denote an orthonormal basis of $U^\perp$ (we do not have this basis readily available, but this is no problem, since, as we shall see, we do not need this basis in the resulting algorithm). Then we can write

$$\Lambda = V_p V_p^T \Lambda + V_r V_r^T \Lambda = V_p \Lambda_p + V_r \Lambda_r,$$

where we use the notation $\Lambda_p = V_p^T \Lambda$ and $\Lambda_r = V_r^T \Lambda$. If we multiply both sides of (6.10) on the right by $(V_p\ V_r)$, where V_r is an orthonormal basis of $U^\perp$, we obtain

$$(\Lambda^T p\ \ \Lambda_r^T) \begin{pmatrix} V_p^T J(x(0))V_p - I & V_p^T J(x(0))V_r \\ 0 & V_r^T J(x(0))V_r - I \end{pmatrix} = (gV_p\ gV_r). \tag{6.11}$$

From equation (6.11) it follows that with the information from the Newton-Picard solution of the state equation, we can immediately solve for $\Lambda_p^T = gV_p(S - I)^{-1}$. So if we have found Λ_p^T, then we have to solve

$$\Lambda^T V_p(V_p^T J(x(0))V_r) + \Lambda^T V_r(V_r^T J(x(0))V_r) - \Lambda^T V_r = gV_r$$

for $\Lambda^T V_r$. This can be done by using the iterative process defined by

$$\lambda_r^{(i+1)} = \lambda_r^{(i)}(V_r^T J(x(0))V_r) + \Lambda^T V_p(V_p^T J(x(0))V_r) - gV_r, \tag{6.12}$$

where $\lambda_r^{(i)}$ is the ith approximation of $\Lambda^T V_r$. Because the Newton-Picard algorithm for the state equations does not provide us with V_r, we would like to avoid the computation

and use of V_r. For this purpose, we note that the iteration (6.12) is equivalent to the iteration

$$(\Lambda^T V_p \;\; \lambda_r^{(i+1)}) = (\Lambda^T V_p \;\; \lambda_r^{(i)}) \begin{pmatrix} V_p^T J(x(0)) V_p & V_p^T J(x(0)) V_r \\ 0 & V_r^T J(x(0)) V_r \end{pmatrix} - (gV_p \; gV_r).$$

If we would start this iteration with $\lambda_r^{(0)} = 0$, then we see that the iteration is nothing else than the iteration

$$\Lambda_{i+1} = J(x(0))^T(\Lambda_i) - g \quad \text{with} \quad \Lambda_0 = V_p \Lambda_p. \tag{6.13}$$

The Newton-Picard method is constructed in such a way that this last iterative procedure converges with rate equal to the modulus of the $(p+1)$th eigenvalue of $J(x(0))$, which is much faster than the Picard iteration of the inverse of the period map K.

The above shows that one could solve (6.10) in theory by first computing $V_p\Lambda_p$ only once and then iterating (6.13) until the desired accuracy is achieved. In practice, however, the iteration (6.13), will only in the beginning converge with rate equal to the modulus of the $(p+1)$th eigenvalue of $J(x(0))$. After a number of iterations, due to round-of errors, the iteration will converge again with rate equal to the modulus of the largest eigenvalue of $J(x(0))$. Therefore it is better not to solve (6.10) directly, but to use a Newton approach. This means iterating

$$\Lambda_{i+1} = \Lambda_i - \Delta\Lambda_i,$$

where $\Delta\Lambda_i$ satisfies

$$(J(x(0)) - I)^T \Delta\Lambda_i = (J(x(0)) - I)^T \Lambda_i - g^T = K^{-1}(\Lambda_i) - \Lambda_i.$$

This equation is basically the same as equation (6.10), but here $K^{-1}(\Lambda_i) - \Lambda_i$ takes the roll of g. Consequently, we can obtain an approximation for $\Delta\Lambda_i$ using the same method as discussed above, but we use now only a few iterations of the scheme (6.13).

The Newton-Picard method for the adjoint equations, as discussed above, can be described with the following four steps. The $\lambda^{(j)}$ in the algorithm are approximations to $\Delta\Lambda_i$, and the Λ_i are approximations to Λ

Newton-Picard Algorithm–Adjoint Equations

Step 1: *Choose an initial value* Λ_0. *Set* $i = 0$

Step 2: *If* $|K^{-1}(\Lambda_i) - \Lambda_i| < \delta_0$ *then stop (1 evaluation of* K^{-1}*), where* K^{-1} *is given by (6.9).*

Step 3: *Compute* $\tilde{\Lambda}^T = (K^{-1}(\Lambda_i) - \Lambda_i)^T V_p (S - I)^{-1}$.

Step 4: *Iterate* l *times:* $\lambda^{(j+1)} = J(x(0))^T \lambda^{(j)} - (K^{-1}(\Lambda_i) - \Lambda_i)$ *with* $\lambda^{(0)} = V_p \tilde{\Lambda}$ *(*l *evaluations of* K^{-1}*). Put* $\Lambda_{i+1} = \Lambda_i - \lambda^{(l)}$. *Set* $i = i + 1$. *Go to* **Step 2**.

The above algorithm amounts to solving $l+1$ initial value problems for each iteration. In our implementation we take $l = 2$. Thus for each iteration of the Newton-Picard method for the adjoint equations we need to solve 3 initial value problems.

6.3 The Optimization of Two Rapid Pressure Swing Processes

In this section we will present the model equations and discuss the optimization of a rapid pressure swing adsorber and of a rapid pressure swing reactor.

6.3.1 The Rapid Pressure Swing Adsorber

The Model Equations

The rapid pressure swing adsorber (RPSA) we consider in this section is the same adsorber as discussed in Chapter 3. In Chapter 3, we only computed a periodic state of the RPSA for a fixed set of parameters. In this chapter, however, we would like to optimize the RPSA with respect to a number of parameters, among which the cycle time. In the model equations as given in Chapter 3, the cycle time appears only implicitly in the definition of the dimensionless parameters. For the purpose of the optimization, it is more convenient to rewrite the model equations in such a way that the physical cycle time appears explicitly as an dimensionless parameter in the equations. In this section we state the rewritten model equations.

The model gas consists of two components, oxygen and nitrogen. The variables in the model are the gas phase concentrations of these two components, denoted by $c^{(i)} : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to c^{(i)}(t,z)$, ($i = 1$: oxygen, $i = 2$: nitrogen), the adsorbed amount of the two components, denoted by $q^{(i)} : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to c^{(i)}(t,z)$, and the gas phase velocity $v : [0,\infty) \times [0,1] \to \mathbb{R},\ (t,z) \to v(t,z)$. Note that the dimensionless cycle time is rescaled to become one time unit. This means the physical cycle time appears as a parameter in the model equations and is denote by T. The model equations consist of the gas phase balance equations

$$c_t^{(i)} = TC_1 c_{zz}^{(i)} - C_2 (v c^{(i)})_z - C_3^{(i)} q_t^{(i)}, \quad i = 1,2, \tag{6.14}$$

the adsorption rate equations

$$q_t^{(i)} = \frac{Tk^{(i)}}{c^{(1)} + c^{(2)} + D^{(i)}} (c^{(i)} - q^{(i)}), \quad i = 1,2, \tag{6.15}$$

and the gas phase velocity equation

$$v(t,z) = -TC_4 \big(c_z^{(1)}(t,z) + c_z^{(2)}(t,z) \big). \tag{6.16}$$

Note that the equations are autonomous, but the boundary conditions are periodic in time. In fact, at the feed end (at $z = 0$) of the adsorber there are two sets of alternating boundary conditions. During the first step, the so-called *pressurization step*, the boundary conditions are

$$\begin{cases} TC_1 c_z^{(i)}(t,0) = C_2 v(t,0)(c^{(i)}(t,0) - y_f^{(i)} p_f), \quad i = 1,2 \\ c^{(1)}(t,0) + c^{(2)}(t,0) = p_f \end{cases} \quad \text{for } 0 \le t \,(\mathrm{mod}\ 1) < t_f. \tag{6.17}$$

Table 6.1: The dimensionless parameter values for the RPSA process

C_1	0.0014	C_2	1.4134
$C_3^{(1)}$	3.8988	$C_3^{(2)}$	8.3975
C_4	0.616	C_5	0.0024
$k^{(1)}$	146.07	$k^{(2)}$	67.82
$D^{(1)}$	0.5351	$D^{(2)}$	0.5007
$y_f^{(1)}$	0.21	$y_f^{(2)}$	0.79
p_w	0.4717	t_f	0.5
a	1.24	b	1.4

During the second step, the so-called *depressurization step*, the boundary conditions are

$$\begin{cases} c_z^{(i)}(t,0) = 0, \quad i = 1,2 \\ c^{(1)}(t,0) + c^{(2)}(t,0) = p_w \end{cases} \quad \text{for } t_f \leq t \,(\text{mod } 1) \leq 1. \tag{6.18}$$

The boundary conditions at the product end (at $z = 1$) of the adsorber are given by

$$\begin{cases} c_z^{(i)}(t,1) = 0, \quad i = 1,2 \\ v(t,1) = TC_5\big(c^{(1)}(t,1) + c^{(2)}(t,1)\big)^{-1} \end{cases} \tag{6.19}$$

The equations (6.14)-(6.16) together with the boundary conditions (6.17)-(6.19) complete the description of the RPSA model. A solution of (6.14)-(6.16) together with the boundary conditions (6.17)-(6.19) that satisfies

$$\begin{cases} c^{(i)}(0,z) = c^{(i)}(1,z), \\ q^{(i)}(0,z) = q^{(i)}(1,z), \end{cases} \quad i = 1,2,$$

is called a periodic solution of the RPSA model.

The dimensionless parameters that appear in the equations are denoted by T, C_1, C_2, $C_3^{(i)}$, C_4, C_5, $k^{(i)}$, $D^{(i)}$, $y_f^{(i)}$, p_w, p_f and t_f. The parameters T (cycle time), p_f (feed pressure) and t_f (ratio between the two step lengths) are used to optimize the process. The values of the other parameters are listed in Table 6.1.

The Performance Index and the Constraints

For the optimization of the RPSA process we will use a measure for the performance of the process, the so-called performance index. For the RPSA process, we would like to minimize the average power requirements of the compressor during the pressurization

step. Thus the performance index to be minimized, is the average power, which is given by

$$V = \frac{P_0}{T} \int_0^{t_f} v(t,0) p_f (a(p_f)^{(b-1)/b} - 1) dt,$$

where P_0 is the characteristic power (for the values of the parameters as given in Table 6.1 we obtain P_0=184.53 [W=J/s]). We wish to optimize this performance index with the constraint that the purity of the product gas is at least 87 %. This purity is given by

$$Pu = \int_0^1 \frac{c^{(1)}(t,1)}{c^{(1)}(t,1) + c^{(2)}(t,1)} dt.$$

Optimization Results

The model equations are discretized on 20 nodes using a second order finite volumes approximation. For the time integration of the resulting system of 80 ($N = 80$) ordinary differential equations, the fortran NAG library routine D02EJF is used.

The first step is to optimize the RPSA with respect to the feed pressure p_f, the duration of the pressurization step t_f, and the duration of the cycle T. The reference case values are given by

$$p_f = 1, \quad t_1 = 0.5, \quad \text{and } T = 3.$$

For these values a purity level of 87.2 % is obtained and the value of the performance index equals 109.4.

For the optimization with respect to these parameters we use the fortran NAG library routine E04UCF (a sequential quadratic programming method). This routine gives the following values for the optimal parameters

$$p_f = 0.998, \quad t_f = 0.201, \quad \text{and } T = 8.38. \tag{6.20}$$

For these values a purity level of 87.0 % is obtained and the corresponding value of the performance index equals 53.9. This is a reduction of more than 50 % with respect to the reference case values.

Since the performance index depends strongly on the feed pressure, the performance index is expected to decrease even further when compared to the three parameter optimization, when the feed pressure is allowed to vary continuously in time.

For the computation of the optimal feed pressure history, we fix the cycle duration T to be equal to 8 and the (dimensionless) duration of the pressurization step to be equal to 0.5. Note that the boundary condition (6.17) needs careful attention. Indeed, the feed pressure may not become so low that the gas velocity at the feed end of the adsorber becomes negative. Therefore, it is more convenient to choose the gas velocity at the entrance of the adsorber as the control parameter instead of the feed pressure. Once the periodic state has been computed, the optimal gas velocity history can be used to obtain the optimal feed pressure history without affecting the periodic state.

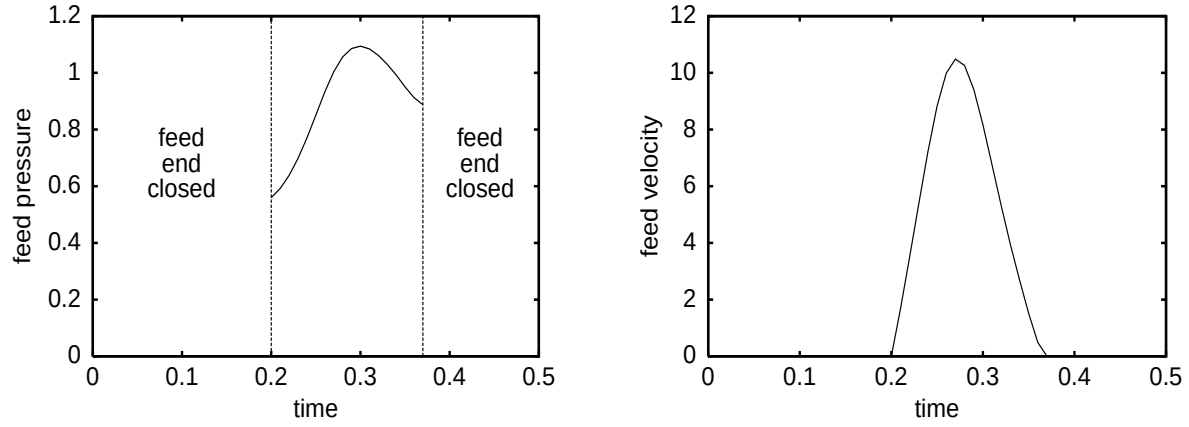

Figure 6.1: The dimensionless feed pressure p_f (left figure) and the dimensionless feed velocity $v(t,0)$ (right figure) versus the dimensionless time t for the optimal time-programmed feed velocity for the RPSA with parameter values as in Table 1. A purity of 87.0 % is obtained and the performance index equals 36.87

The resulting optimal gas velocity and feed pressure histories are given in Fig. 6.1. For this feed pressure history, the purity turns out to be 87.0 % and the performance index is 36.87. This is a considerable improvement compared to the solution of the minimization using only constant parameters.

From the graph of the gas velocity history in Fig. 6.1, it is seen that at the beginning and at the end of the pressurization step, the gas velocity is equal to zero for some time. This situation can also be accomplished by closing the feed end of the adsorber. This suggests that, even when it is not possible to control the feed pressure continuously, the performance index can still be decreased by incorporating two more steps in the cycle scheme. This cycle scheme can be described by the following sets of boundary conditions. During the first step, *the pre-pressurization step*, the boundary conditions are

$$\begin{cases} c_z^{(i)}(t,0) = 0, \quad i = 1,2 \\ v(t,0) = 0 \end{cases} \quad \text{for } 0 \le t \,(\text{mod } 1) < t_1.$$

During the second step, *the pressurization step*, the boundary conditions are

$$\begin{cases} TC_1 c_z^{(i)}(t,0) = C_2 v(t,0)(c^{(i)}(t,0) - y_f^{(i)} p_f), \quad i = 1,2 \\ c^{(1)}(t,0) + c^{(2)}(t,0) = p_f \end{cases} \quad \text{for } t_1 \le t \,(\text{mod } 1) < t_1 + t_2.$$

During the third step, *the post-pressurization step*, the boundary conditions are the same as in the first step:

$$\begin{cases} c_z^{(i)}(t,0) = 0, \quad i = 1,2 \\ v(t,0) = 0 \end{cases} \quad \text{for } t_1 + t_2 \le t \,(\text{mod } 1) < t_1 + t_2 + t_3.$$

During the fourth step, *the depressurization step*, the boundary conditions are

$$\begin{cases} c_z^{(i)}(t,0) = 0, \quad i = 1,2 \\ c^{(1)}(t,0) + c^{(2)}(t,0) = p_w \end{cases} \quad \text{for } t_1 + t_2 + t_3 \le t \,(\text{mod } 1) \le 1.$$

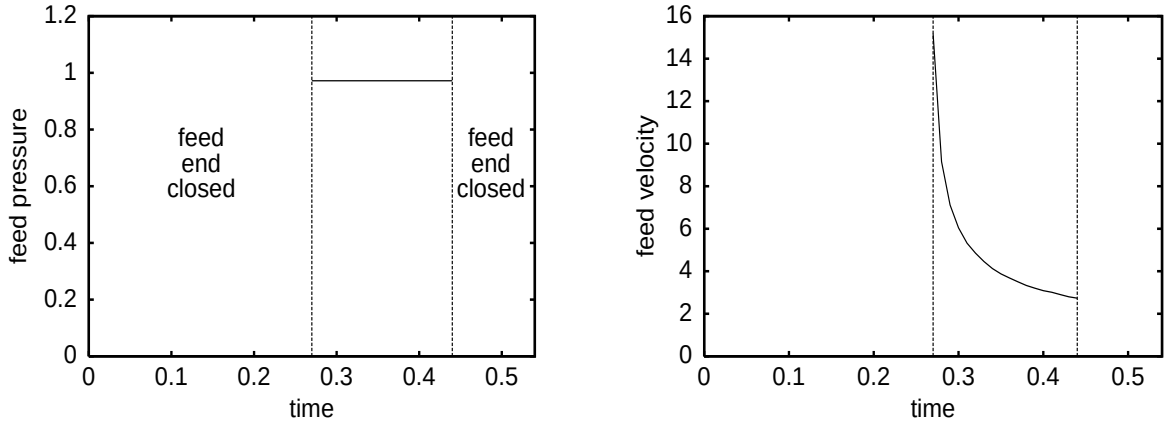

Figure 6.2: The dimensionless feed pressure p_f (left figure) and the dimensionless feed velocity $v(t,0)$ (right figure) versus the dimensionless time t for the optimal four step cycle for the RPSA with parameter values as in Table 1. A purity of 87.0 % is obtained and the performance index equals 41.0.

Note that in this step the boundary conditions are again the same as for the original depressurization step.

If we now minimize the performance index with respect to the parameters t_1, t_2, t_3, t_4, p_f and T, again using the fortran NAG library routine E04UCF, the following values are obtained

$$t_1 = 0.263, \quad t_2 = 0.179, \quad t_3 = 0.097, \quad T = 7.73, \quad \text{and } p_f = 0.972.$$

The corresponding purity is 87.0% and the value of the performance index equals 41.0. This is approximately a 25% improvement compared to the three parameter minimization (6.20). The corresponding cycle scheme is depicted in Fig. 6.2.

6.3.2 The Rapid Pressure Swing Reactor

The Model Equations

This second process that we consider, consists of a single bed, which is packed with a mixture of an active catalyst for the reaction, and a selective adsorbent for the adsorption of one or more of the reaction species. The process is operated with two basic steps: feed gas pressurization and counter-current depressurization. In the pressurization step, feed is introduced at a pressure higher than atmospheric pressure. Adsorption and reaction occur simultaneously within the reactor. Different component adsorption capacities promote separation of the reactants from the products and thus may increase conversion of the reactant and/or selectivity of the desired product. In the depressurization step, the pressure in the bed is reduced to atmospheric pressure, the adsorbed species are desorbed and the bed is subsequently regenerated. Both steps have a short time duration, in the order of several seconds. The adsorption bed consists of small adsorbent particles with average size between 200-700 μm. This combination of a fast cycle time together with a

small particle size leads to steep and periodically varying pressure gradients within the bed. The pressure at the product end is approximately constant over time and this makes the process useful for continuous product release.

A reversible dissociation reaction, $2\mathrm{A} \rightleftharpoons \mathrm{B} + \mathrm{C}$, is investigated. The model gas consists of three components, A, B and C. The variables in the model are the gas phase concentrations of these three components, denoted by $c^{(i)} : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to c^{(i)}(t,z)$, (with $i = 1$: A, $i = 2$: B and $i = 3$: C), the adsorbed amount of the three components, denoted by $q^{(i)} : [0,\infty) \times [0,1] \to [0,\infty),\ (t,z) \to c^{(i)}(t,z)$, and the gas phase velocity $v : [0,\infty) \times [0,1] \to \mathbb{R},\ (t,z) \to v(t,z)$. Note that the dimensionless cycle time is rescaled to become one time unit. This means the physical cycle time appears as a parameter in the model equations and is denote by T. The model equations consist of the gas phase balance equations

$$c_t^{(i)} = TC_1 c_{zz}^{(i)} - C_2(vc^{(i)})_z - C_3^{(i)} q_t^{(i)} - Tk^{(i)} r(c^{(1)}, c^{(2)}, c^{(3)}), \quad i = 1,2,3, \tag{6.21}$$

the adsorption rate equations

$$q_t^{(i)} = c_t^{(i)}, \quad i = 1,2,3, \tag{6.22}$$

the gas phase velocity equation

$$v(t,z) = -TC_4\big(c_z^{(1)}(t,z) + c_z^{(2)}(t,z) + c_z^{(3)}(t,z)\big), \tag{6.23}$$

and the reaction rate equation

$$r\big(c^{(1)}(t,z), c^{(2)}(t,z), c^{(3)}(t,z)\big) = \left(c^{(1)}(t,z)^2 - \frac{c^{(2)}(t,z)c^{(3)}(t,z)}{K}\right). \tag{6.24}$$

Note that the equations are autonomous, but that the boundary conditions are periodic in time. In fact, at the feed end (at $z = 0$) of the reactor there are two sets of alternating boundary conditions. During the first step, the so-called *pressurization step*, the boundary conditions are

$$\begin{cases} TC_1 c_z^{(i)}(t,0) = C_2 v(t,0)(c^{(i)}(t,0) - y_f^{(i)} p_f), \quad i = 1,2,3 \\ c^{(1)}(t,0) + c^{(2)}(t,0) + c^{(3)}(t,0) = p_f \end{cases} \quad \text{for } 0 \le t \,(\mathrm{mod}\ 1) < t_f. \tag{6.25}$$

During the second step, the so-called *depressurization step*, the boundary conditions are

$$\begin{cases} c_z^{(i)}(t,0) = 0, \quad i = 1,2,3 \\ c^{(1)}(t,0) + c^{(2)}(t,0) + c^{(3)}(t,0) = p_w \end{cases} \quad \text{for } t_f \le t \,(\mathrm{mod}\ 1) \le 1. \tag{6.26}$$

The boundary conditions at the product end (at $z = 1$) of the reactor are given by

$$\begin{cases} c_z^{(i)}(t,1) = 0, \quad i = 1,2,3 \\ v(t,1) = TC_5\big(c^{(1)}(t,1) + c^{(2)}(t,1) + c^{(3)}(t,1)\big)^{-1} \end{cases} \tag{6.27}$$

Table 6.2: The dimensionless parameter values for the RPSR process

C_1	$1.40{\cdot}10^{-5}$	C_2	1.40
$C_3^{(1)}$	0	$C_3^{(2)}$	0
$C_3^{(3)}$	1.163	C_4	1.064
C_5	0.000170	$k^{(1)}$	36198
$k^{(2)}$	-18099	$k^{(3)}$	-18099
K	0.0156	$y_f^{(1)}$	1
$y_f^{(2)}$	0	$y_f^{(3)}$	0
p_f	1	p_w	0.33
t_f	0.5		

The equations (6.21)-(6.24) together with the boundary conditions (6.25)-(6.27) complete the description of the RPSR model. A solution of (6.21)-(6.24) together with the boundary conditions (6.25)-(6.27) that satisfies

$$\begin{cases} c^{(i)}(0,z) = c^{(i)}(1,z), \\ q^{(i)}(0,z) = q^{(i)}(1,z), \end{cases} \quad i = 1,2,3$$

is called a periodic solution of the RPSR model. The dimensionless parameters that appear in the equations are denoted by T, C_1, C_2, $C_3^{(i)}$, C_4, C_5, $k^{(i)}$, $m^{(i)}$, $y_f^{(i)}$, K, p_w, p_f and t_f. The parameters T (cycle time), p_f (feed pressure) and t_f (ratio between the two step lengths) are used to optimize the process. The values of the other parameters are listed in Table 6.2.

The Performance Index and the Constraints

Also for the optimization of the RPSR process we need a measure for the performance of the process. For the RPSR process, we would like to optimize the yield of the pure product B. Thus the performance index to be maximized, is the yield of the component B, which is given by

$$Y = \frac{\int_0^1 v(t,1)c^{(2)}(t,1)dt}{\int_0^{t_f} v(t,0)c^{(1)}(t,0)dt}.$$

We wish to optimize this performance index with the constraint that the purity of the product gas is at least 95 %. The purity of the product gas is given by

$$Pu = \int_0^1 \frac{c^{(2)}(t,1)}{c^{(1)}(t,1) + c^{(2)}(t,1) + c^{(3)}(t,1)}dt.$$

Optimization Results

We discretize the model equations on 20 nodes using a third order upwind finite volumes approximation. For the time integration of the resulting system of 60 ($N = 60$) ordinary differential equations, the fortran NAG library routine D02EJF is used. Here we take as control parameters the feed pressure p_f, the duration of the pressurization step t_f, and the duration of the cycle T. The reference case values are given by

$$p_f = 1, \quad t_f = 0.5, \quad \text{and } T = 4.5.$$

For these values a purity level of 92.31 % is obtained and the corresponding value of the performance index equals 0.0014. Note that these parameters do not satisfy the constraint on the purity.

We use the fortran NAG library routine E04UCF (a sequential quadratic programming method) to find the optimal values of the parameters that do satisfy the purity constraint. This routine gives the following values for the optimal parameters

$$p_f = 1.12474, \quad t_f = 0.431, \quad \text{and } T = 4.46567.$$

For these values a purity level of 94.9922 % is obtained and value of the performance index equals 0.0012.

It is expected that, when the feed pressure is allowed to vary continuously in time, the performance index can be increased even further in comparison with the previous three parameter optimization.

For the computation of the optimal feed pressure history, we fix the cycle duration T to 4.41 and the (dimensionless) duration of the pressurization step to 0.5. We again have to pay attention to the boundary condition (6.25). For the same reason as for the RPSA process, also for the RPSR the gas velocity at the entrance of the reactor is used as the control parameter, rather than the feed pressure.

The resulting optimal gas velocity and feed pressure histories are given in Fig. 6.3. For this feed pressure history, the purity turns out to be 95.0 % and the performance index is 0.00339. This is a considerable improvement compared to the solution of the optimization using only time-independent parameters.

From the graph of the gas velocity history in Fig. 6.3 it is seen that at the beginning and at the end of the pressurization step the gas velocity is equal to zero for some time. This means that also for the RPSR, the performance index may still be decreased by incorporating two more steps in the cycle scheme. This four-step cycle scheme can be described by the following sets of boundary conditions. During the first step, *the pre-pressurization step*, the boundary conditions are

$$\begin{cases} c_z^{(i)}(t,0) = 0, \quad i = 1,2,3 \\ v(t,0) = 0 \end{cases} \quad \text{for } 0 \leq t \,(\text{mod } 1) < t_1.$$

During the second step, *the pressurization step*, the boundary conditions are

$$\begin{cases} TC_1 c_z^{(i)}(t,0) = C_2 v(t,0)(c^{(i)}(t,0) - y_f^{(i)} p_f), \quad i = 1,2,3 \\ c^{(1)}(t,0) + c^{(2)}(t,0) + c^{(3)}(t,0) = p_f \end{cases} \quad \text{for } t_1 \leq t \,(\text{mod } 1) < t_1 + t_2.$$

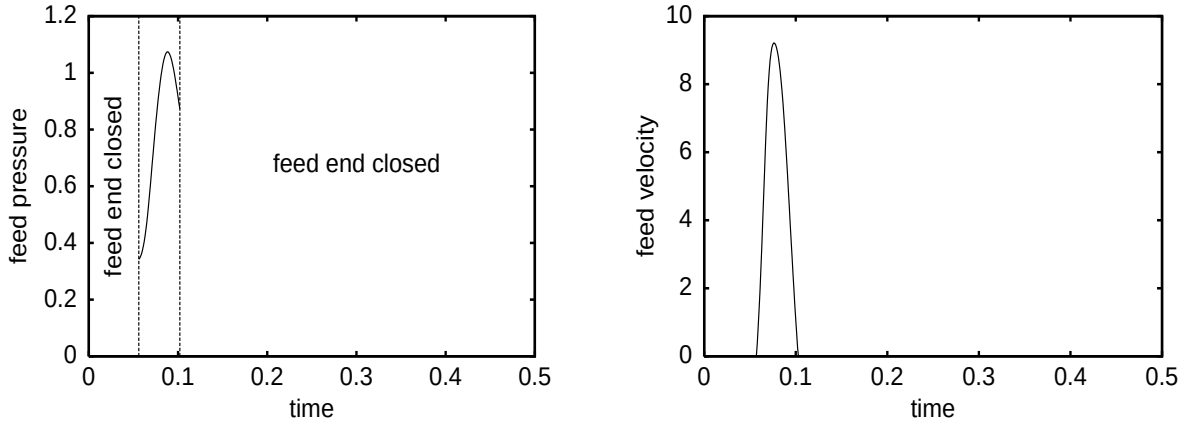

Figure 6.3: The dimensionless feed pressure p_f (left figure) and the dimensionless feed velocity $v(t,0)$ (right figure) versus the dimensionless time t for the optimal time-programmed feed velocity for the RPSR with parameter values as in Table 2. A purity of 95.0 % is obtained and the performance index equals 0.00339.

During the third step, *the post-pressurization step*, the boundary conditions are the same as in the first step:

$$\begin{cases} c_z^{(i)}(t,0) = 0, \quad i = 1,2,3 \\ v(t,0) = 0 \end{cases} \quad \text{for } t_1 + t_2 \leq t \,(\text{mod } 1) < t_1 + t_2 + t_3.$$

During the fourth step, *the depressurization step*, the boundary conditions are

$$\begin{cases} c_z^{(i)}(t,0) = 0, \quad i = 1,2,3 \\ c^{(1)}(t,0) + c^{(2)}(t,0) + c^{(3)}(t,0) = p_w \end{cases} \quad \text{for } t_1 + t_2 + t_3 \leq t \,(\text{mod } 1) \leq 1.$$

Note that in this step the boundary conditions are again the same as for the original depressurization step.

If we now optimize the performance index with respect to the parameters t_1, t_2, t_3, p_f and T, again using the fortran NAG library routine E04UCF, the following values are obtained

$$t_1 = 0.0286, \quad t_2 = 0.0282, \quad t_3 = 0.301, \quad T = 4.485, \quad \text{and } p_f = 1.000.$$

The corresponding purity is 95.06% and the value of the performance index equals 0.00338. This optimum is very close to the optimal control and it improves the performance index by approximately 300% compared to the three parameter optimization (6.28). The corresponding cycle scheme is depicted in Fig. 6.4

Thus we see that, for both the RPSA and the RPSR, even in the case it is not possible to vary the control parameter continuously in time, it is fruitful to consider the solution to the optimal control problem. From this solution, insight can be obtained on how to change the cycle scheme in order to improve the performance of the process. In case of the studied rapid pressure swing processes, it is observed that the incorporation of

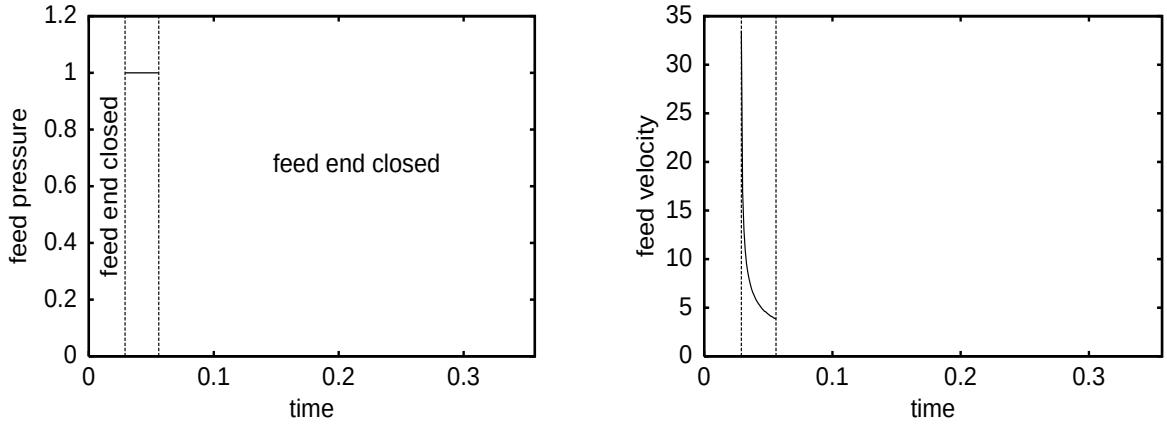

Figure 6.4: The dimensionless feed pressure p_f (left figure) and the dimensionless feed velocity $v(t,0)$ (right figure) versus the dimensionless time t for the optimal four step cycle for the RPSR with parameter values as in Table 2. A purity of 95.06 % is obtained and the performance index equals 0.00338.

two additional steps in the cycle scheme greatly improves the performance of the process using only time-independent control parameters. We therefore believe that the numerical procedure to compute solutions to the optimal control problem, that we have used to obtain the results in this section and that we present in the next section, can be an important tool in the study of periodic processes.

6.4 Notes

The contents of this chapter are based on [60].

A good description and derivation of the used first order gradient method can be found in [5]. In this reference also the role of the two damping factors ϵ_1 and ϵ_2 is explained.

Basically, we are using a similar optimal control technique as used by [17] and [27]; these authors use a combination of a first order gradient method with Newton's method. Due to the rapid increase in computer time as the dimension of the system grows, in these two papers, this technique was only applied to low dimensional systems. The interaction with the Newton-Picard method makes the first order gradient method much more efficient. Therefore we can apply the first order gradient method to higher dimensional systems and thus we are able to make much more accurate computations.

Other work on the optimization of periodic processes can be found in [31], where a trace separation system is optimized. In [32], the same authors consider the efficient determination of periodic states of cyclic processes. In [38] a complete discretization approach is used to determine periodic states and optimize two periodic adsorption processes. [52] employ quasi-Newton methods for the efficient optimization of a continuous adsorption process in a dessicant rotor.

The model equations and parameter values for the rapid pressure swing adsorber are taken from [38] and the model equations and parameter values for the rapid pressure swing reactor are taken from [6].

Chapter 7

Conclusions

In this thesis we have presented methods and algorithms for the efficient computation of periodic states of cyclically operated chemical processes. We have observed in Chapter 3, where the model equations for a rapid pressure swing adsorber and a reverse flow reactor are presented, that these processes exhibit low dimensional dynamics, that the leading eigenvalues of the Jacobian of these systems are real and that especially the smaller eigenvalues do not vary much for different parameter values and different states. These properties are exploited by the Newton-Picard method that we have presented in this thesis.

The Newton-Picard method is new in chemical engineering literature and is a variant of the method described in [37]. There are some differences between our approach and the one in [37]. One of them is that we keep the dimension of the subspace U, the part of the state space $\mathbb{R}^N$ on which the action of the Jacobian is computed, fixed. Another difference is that we perform the subspace iterations, which update the basis of the subspace U, only on p (the dimension of U) vectors. These differences are motivated by the above listed properties of the cyclic chemical processes. The presented method is specifically intended to be used for the kind of processes discussed in this thesis. For systems for which the eigenvalues are complex and change rapidly, the approach in [37] should be used.

In Chapter 3, we have applied the method to the rapid pressure swing adsorber and the reverse flow reactor. For these systems it turns out that the Newton-Picard method is much more efficient than either a Picard method or a Newton's method. The Newton-Picard method is also able to compute accurately the largest eigenvalues of the Jacobian of the period map evaluated in the periodic solutions. The method is therefore able to determine the stability of these solutions.

In Chapter 4, we have compared the performance of Newton-Picard to the performance of three existing methods, dynamic simulation, Newton's method and Broyden's method. We have applied all four the methods to five basic models of periodic processes that are well documented in chemical engineering literature. Two different convergence criteria are introduced, one based on the residual of a state and the other based on the deviation of a state. For all the test cases the ordering of the methods from the most efficient to the least efficient turned out to be independent of the convergence criterion used.

Two of the presented systems (the H_2O/air and CO_2/He PSA separation) exhibit linear behaviour. These two systems differ in the size of the capacity terms. This difference explains why the dynamic simulation of the CO_2/He system converges much faster to a periodic state than the dynamic simulation of the H_2O/air system. It is shown that for linear systems the largest eigenvalue of the Jacobian of the period map provides a good indication for the convergence rate of the dynamic simulation. Using this information, the number of cycles needed for a dynamic simulation to converge can be determined a priori. Therefore, it can be calculated whether or not Newton's method is more efficient with respect to the number of cycle simulations. For linear systems, Broyden's method is known to converge in at most twice the convergence time of Newton's method, but the two test systems show that actually Broyden's method converges much faster. It is also shown that, if we find a stable periodic state for a linear system, it necessarily follows that the periodic state is unique.

It is found that the Newton-Picard method is in most cases much more efficient than dynamic simulation or Newton's method. Only for the weakly nonlinear CO_2/N_2 PSA separation, the Newton-Picard method needs approximately the same number of evaluations of the period map as the dynamic simulation, which converged relatively fast. In addition the Newton-Picard method is able to converge to unstable periodic states and is able to compute the dominant eigenvalues of a periodic state.

Broyden's method is in the tested cases always the most efficient in terms of needed cycle simulations. However, it is shown that Broyden can suffer from robustness problems in nonlinear cases. It is also shown that the comparison of Broyden vs. Newton-Picard is not a fair one. The Newton-Picard method gives more information about the studied systems than Broyden's method, as the Newton-Picard method also computes approximations to the eigenvalues that determine the stability of the computed periodic states. Thus in situations where the stability of periodic states is needed the Newton-Picard method is clearly superior to Broyden's method.

It is seen that the reported "good" value of 0.5 for the parameter ρ in the Newton-Picard method [37] is not always the best choice. It actually might happen that that the Newton-Picard method does not converge with $\rho = 0.5$ (see the H_2S/natural gas PSA system). In this case ρ has to be chosen smaller in order to have convergence.

In Chapter 5, we have introduced a new method for the efficient computation of branches of periodic solutions of periodically forced PDE's. We refer to this method as the Broyden rank $p + 1$ update continuation method with Subspace Iteration, or, in short, the BSI rank $p + 1$ method. The method makes efficiently use of the central ideas behind Broyden's method and behind the Newton-Picard methods developed by [37]. As the name suggests, the method combines a Broyden method with a subspace iteration procedure for the computation of invariant subspaces. Simultaneously approximations to a periodic solution and to the eigenvalues that determine the stability are computed. In this way bifurcation points can be detected accurately and efficiently.

We have used the method to compute a branch of periodic states of a cooled reverse flow reactor. We have also used two variants of the Newton-Picard method [37] to compute

the same branch. For the example system, the BSI rank $p+1$ method proved to be the most efficient method, in terms of number of period map evaluations. In fact, the BSI method needed approximately half the number of evaluations of F as the next most efficient method (CNP with $l = 2$ and $p = 7$). The main advantage of the BSI rank $p+1$ method, when compared to the Newton-Picard methods, is the small number of period map evaluations per iteration of the method. The method proved also to have the largest domain of attraction. This results in the largest average step size, so that the method needed the fewest continuation points.

Of the two variants of the Newton-Picard method, the CNP and the CNPGS variant, the CNPGS is the least expensive variant in terms of number of evaluations of F per iteration. The CNP method, however, is also the least robust method of the two, and this results in the poorest overall performance.

We showed that the performance of the BSI rank $p+1$ method in terms of number of evaluations of F is essentially independent of the number of nodes used in the discretization. This fact is already known for the Newton-Picard methods, see [37].

We believe that a more sophisticated variant of the subspace iteration with locking and deflation and an adaptive control of the dimension of the subspace on which the subspace iteration is performed, such as used in the methods of [37], can also be applied in the BSI rank $p+1$ method. This will increase the efficiency of the new method even further.

In Chapter 6, we introduced an efficient numerical optimization procedure for the computation of solutions to periodic optimal control problems. This procedure consists of a combination of the Newton-Picard method with a first order gradient method. This latter method is able to compute the optimal control of a periodic process under one (or more) nonlocal constraints, such as a minimum product purity constraint. The Newton-Picard method does not only compute periodic solutions to the state equations, but also obtains information about the stability of the computed solution. This information is used by the method to compute, with a small additional computational effort, periodic solutions to the adjoint equations, needed in the first order gradient algorithm.

Using the presented numerical procedure, we have studied the numerical optimization of two periodic processes: a rapid pressure swing adsorber [38] and a rapid pressure swing reactor [6]. For both processes we presented the model equations.

First, the two processes were optimized with respect to three time-constant parameters: the cycle duration, the feed pressure, and the ratio between the two steps in the cycle. Subsequently, we computed for both processes the optimal control of the feed pressure during the pressurization step, using the first order gradient method in combination with the Newton-Picard method. The computed optimal controls improved the performance of the processes considerably, compared to the time-constant parameter optima.

The computed optimal controls suggested that the addition of two more steps in the operating scheme, a pre- and a post-pressurization step in which the feed end of the adsorber or reactor is closed, would also improve the time-constant parameter control performance of the processes. To demonstrate this, we optimized the two processes with

respect to five time-constant parameters: the duration of the four steps in the cycle, and the feed pressure during the pressurization step. The found four step cycle optima improved the performance of the optimal two step cycles with approximately 25%. This result extends the result by [27], who found that the optimal control of a period adsorber consists of a three step cycle: a pressurization step, a post-pressurization step with no inflow, and a depressurization step.

As a last remark, we would like to point out that the work in Chapter 6 shows that the results of a time-dependent parameter optimization can help to gain insight in the solution of a time-constant parameter optimization problem. Also for this reason the optimal control method presented in Chapter 6 will prove to be a invaluable tool for the study and design of periodic processes.

Appendix A

Parameter Values

A.1 Parameters for the Rapid Pressure Swing Adsorber

The dimensionless parameters for the rapid pressure swing adsorber in terms of the physical parameters are given by

$$\begin{aligned}
C_1 &= D_{ax}T/(L^2(\varepsilon_b + \varepsilon_p(1-\varepsilon_b))),\\
C_2 &= 1/(\varepsilon_b + \varepsilon_p(1-\varepsilon_b)),\\
C_3^{(i)} &= \rho_b m_i R T_0/(\varepsilon_b + \varepsilon_p(1-\varepsilon_b)),\\
C_4 &= T d_p^2 \varepsilon_b^3 P_f/(180\mu(1-\varepsilon_b)^2 L^2),\\
C_5 &= Q P_{atm} T/(A P_f L),\\
D^{(i)} &= D_m/D_k^{(i)},\\
k^{(i)} &= 60 T \varepsilon_p^2 (1-\varepsilon_b) D_m/(d_p^2 \rho_b m_i R T_0 \tau_p),\\
p_w &= P_{atm}/P_f
\end{aligned}$$

The values of the physical parameters are given in Table A.1.

Table A.1: The physical parameter values for the RPSA process. Super- or subscript 1 denotes oxygen, 2 denotes nitrogen.

D_{ax}	10^{-3} m^2/s	axial dispersion coefficient
D_m	$9.376{\cdot}10^{-6}$ m^2/s	molecular diffusion coefficient
m_1	$3.08{\cdot}10^{-6}$ m^2 mol/(N kg)	adsorption isotherm coefficient
m_2	$1.43{\cdot}10^{-6}$ m^2 mol/(N kg)	adsorption isotherm coefficient
P_{atm}	$1.013{\cdot}10^5$ N/m^2	atmospheric pressure
P_f	$2.148{\cdot}10^5$ N/m^2	feed pressure
d_{pore}	$0.12{\cdot}\ 10^{-6}$ m	pore diameter
$y_f^{(1)}$	0.21 (-)	feed oxygen mole fraction
$y_f^{(2)}$	0.79 (-)	feed nitrogen mole fraction
ε_p	0.55 (-)	adsorbent void fraction
ε_b	0.35 (-)	bed void fraction
Q	10^{-5} m^3/s	product delivery rate
ρ_b	800 kg/m^3	bed bulk density
μ	$1.8{\cdot}\ 10^{-5}$ N s/m^2	gas viscosity
τ_p	3 (-)	particle tortuosity factor
L	1 m	bed length
d_p	$3.025{\cdot}10^{-4}$ m	particle size
T	3 s	cycle time
$D_k^{(1)}$	$1.752{\cdot}10^{-5}$ m^2/s	Knudsen diffusion coefficient
$D_k^{(2)}$	$1.873{\cdot}10^{-5}$ m^2/s	Knudsen diffusion coefficient
T_0	290 K	feed temperature
A	$1.9662{\cdot}10^{-3}$ m^2	bed cross sectional area
R	8.3145 J/(mol K)	universal gas constant
t_f	0.5 (-)	pressurization duration

A.2 Parameters for the Cooled Reverse Flow Reactor

The dimensionless parameters for the cooled reverse flow reactor in terms of the physical parameters are given by

$$
\begin{aligned}
K_1 &= \frac{t_f((1-\varepsilon)\lambda_s + u^2(\rho c_p)_g^2/(ha_v))}{L^2((\rho c_p)_s(1-\varepsilon) + (\rho c_p)_g\varepsilon)} \\
K_2 &= \frac{t_f u(\rho c_p)_g}{L((\rho c_p)_s(1-\varepsilon) + (\rho c_p)_g\varepsilon)} \\
K_3 &= \frac{\Delta T_{ad} t_f k_\infty e^{E_a/(RT_0)}(\rho c_p)_g}{T_0((\rho c_p)_s(1-\varepsilon) + (\rho c_p)_g\varepsilon)} \\
K_4 &= \frac{2Ut_f}{r((\rho c_p)_s(1-\varepsilon) + (\rho c_p)_g\varepsilon)} \\
K_5 &= \frac{t_f D_{ax}}{L^2} \\
K_6 &= \frac{t_f u}{\varepsilon L} \\
K_7 &= \frac{t_f k_\infty e^{E_a/(RT_0)}}{\varepsilon} \\
g(\theta) &= \frac{a_v k_c e^{E_a(\theta-1)/(\theta RT_0)}}{a_v k_c + k_\infty e^{-E_a/(\theta RT_0)}}
\end{aligned}
$$

The values of the physical parameters are given in Table A.2.

Table A.2: The physical parameter values for the cooled reverse flow reactor

D_{ax}	$4{\cdot}10^{-5}$ m^2s^{-1}	axial diffusion constant
t_f	60 s	flow reversal time
λ_s	1.26 W/(m K)	solid conductivity
L	1 m	reactor length
$(\rho c_p)_g$	624.4 J/(m^3 K)	volumetric heat capacity
$(\rho c_p)_s$	$1.319{\cdot}10^6$ J/(m^3 K)	volumetric heat capacity
u	2 m/s	superficial gas velocity
ΔT_{ad}	50 K	adiabatic temperature rise
ε	0.69 (-)	bed void-age
T_0	323 K	feed temperature
h	130 W/(m^2 K)	fluid/catalyst heat transfer coefficient
r	(varied) m	outer reactor radius
k_∞	$1.815{\cdot}10^7$ s^{-1}	frequency factor
E_a	1001.7 J/mol	activation energy
a_v	2628.8 m^{-1}	geometrical surface area of catalyst particles
k_c	0.115 m/s	mass transfer coefficient
R	8.3145 J/(mol K)	universal gas constant
U	(varied) W/(K m^2)	overall heat transfer coefficient

A.3 Parameters for the CO_2/N_2 and the H_2S/Natural Gas PSA Systems

The physical parameters for the CO_2/N_2 and H_2S/natural gas PSA systems are given in Tables A.3, A.4 and A.5 and their relations to the dimensionless parameters for the CO_2/N_2 PSA system are given by

$$\begin{aligned}
C &= c_{\mathrm{pg}}/R, \\
H_i &= h_i/(R\Theta), \\
k &= k_{\mathrm{ads}}T, \\
K_1 &= TD_{\mathrm{ax}}/L^2, \\
K_2 &= \rho_{\mathrm{b}}\Theta QR/(\epsilon P_{\mathrm{H}}), \\
K_3 &= \rho_{\mathrm{b}}\Theta c_{\mathrm{ps}}/(\epsilon P_{\mathrm{H}}), \\
K_4 &= \lambda T\Theta/(P_{\mathrm{H}}L^2), \\
K_5 &= 2h\Theta T/(rP_H\epsilon), \\
\kappa_1 &= q_{\mathrm{m}}/Q, \\
\kappa_2 &= b_0\Theta^{\alpha-1}P_{\mathrm{H}}/R, \\
\kappa_3 &= Q_A/(R\Theta).
\end{aligned}$$

The relations between the physical parameters and the dimensionless parameters for the H_2S/natural gas PSA system are given by

$$\begin{aligned}
C &= c_{\mathrm{pg}}/R, \\
H_i &= h_i/(R\Theta), \\
\hat{k}_1^{(i)} &= k_1^{(i)}/Q, \\
\hat{k}_2^{(i)} &= k_2^{(i)}\Theta/Q, \\
\hat{k}_3^{(i)} &= k_3^{(i)}P_{\mathrm{H}}, \\
\hat{k}_4^{(i)} &= k_4^{(i)}/\Theta, \\
K_1 &= TD_{\mathrm{ax}}/L^2, \\
K_2 &= \rho_{\mathrm{b}}\Theta QR/(\epsilon P_{\mathrm{H}}), \\
K_3 &= \rho_{\mathrm{b}}\Theta c_{\mathrm{ps}}/(\epsilon P_{\mathrm{H}}), \\
K_4 &= \lambda T\Theta/(P_{\mathrm{H}}L^2), \\
K_5 &= 2h\Theta T/(rP_H\epsilon),
\end{aligned}$$

Table A.3: The Langmuir parameter values for the CO_2/N_2 PSA system

compo-nent	q_m mol/kg	b_0 $m^3/mol\,K$	α -	Q_C J/mol	k_{ads} 1/s
CO_2	2.98	$3.74{\cdot}10^{-20}$	5.4	34230	$9{\cdot}10^{-3}$

Table A.4: The Langmuir parameter values for the H_2S/natural gas PSA system

compo-nent	$k_1^{(i)}$ mol/kg	$k_2^{(i)}$ mol/kg K	$k_3^{(i)}$ Pa^{-1}	$k_4^{(i)}$ K	h_i J/mol
H_2S	15.0364	$3.37{\cdot}10^{-2}$	$1.47{\cdot}10^{-3}$	198.767	$3.276{\cdot}10^4$
CO_2	11.1370	$2.16{\cdot}10^{-2}$	$2.428{\cdot}10^{-7}$	1196.25	$2.502{\cdot}10^4$
CH_4	2.6396	$8.4{\cdot}10^{-5}$	$2.398{\cdot}10^{-9}$	2075.74	$1.385{\cdot}10^4$

Table A.5: The physical parameter values for the CO_2/N_2 and H_2S/natural gas PSA systems

	CO_2/N_2	H_2S/nat. gas		CO_2/N_2	H_2S/nat. gas
ϵ	0.4	0.4	L	1.2 m	5 m
t_{pre}	360 s	1800 s	r	0.0135 m	0.5 m
t_{ads}	360 s	300 s	T	1440 s	2700 s
t_{blo}	360 s	300 s	Θ	293 K	298 K
t_{pur}	360 s	300 s	Q	1 mol/kg	1 mol/kg
u_{ads}	0.2 m/s	0.8 m/s	P_H	4.0 atm	30 atm
u_{pur}	-0.6 m/s	-6.4 m/s	P_L	1.0 atm	1.0 atm
D_{ax}	$3.8{\cdot}10^{-4}$ m^2/s	$3.8{\cdot}10^{-4}$ m^2/s	λ	22 W/m K	22 W/m K
ρ_b	720 kg/m^3	720 kg/m^3	c_{ps}	924 J/kg K	1171.5 J/kg K
R	8.3145 J/mol K	8.3145 J/mol K	h	45 $J/m^2s\,K$	-
c_{pg}	29.1 J/mol K	37.2 J/mol K	n	2	3
t_s	-	120 s	$y_f^{(1)}$	0.1 (CO_2)	0.001 (H_2S)
$y_f^{(2)}$	0.9 (N_2)	0.05 (CO_2)	$y_f^{(3)}$	-	0.949 (CH_4)

A.4 Parameters for the H_2O/air and the CO_2/He PSA Systems

The physical parameters for the H_2O/air and CO_2/He PSA systems are given in Table A.6.

Table A.6: The physical parameter values for the H_2O/air and CO_2/He PSA systems

	H_2O/air	CO_2/He		H_2O/air	CO_2/He
P_H	5.0 atm	4 atm	v_{ads}	0.25 m/s	0.1 m/s
P_L	1.0 atm	1.3 atm	v_{pur}	-0.5 m/s	-0.2 m/s
t_{pre}	30 s	20 s	L	0.5 m	1.1 m
t_{ads}	270 s	180 s	K	9084	52.7
t_{blo}	30 s	20 s	k_{ads}	$2.583 \cdot 10^{-4}$ s^{-1}	$4.67 \cdot 10^{-2}$ s^{-1}
t_{pur}	270 s	180 s	ϵ	0.40	0.42
T	600 s	400 s			

A.5 Parameters for the Reverse Flow Reactor (Chap. 4)

The dimensionless parameters for the reverse flow reactor discussed in Chapter 4 in terms of the physical parameters are given by

$$\begin{aligned} k^* &= k_\mathrm{r} L/U, \\ E^* &= E/(R\Delta T_\mathrm{ad}), \\ \theta^* &= \Theta/\Delta T_\mathrm{ad}, \\ Pe_\mathrm{c} &= UL/D_\mathrm{ax}, \\ Pe_\mathrm{t} &= \epsilon U L \rho_\mathrm{g} c_\mathrm{pg}/\lambda. \end{aligned}$$

The physical parameters are given in Table A.7.

Table A.7: The physical parameters for the reverse flow reactor

Θ	573 K	D_ax	$2{\cdot}10^{-3}$ m^2/s
U	1 m/s	E/R	8000 K
$\rho_\mathrm{g} c_\mathrm{pg}$	500 J/m^3 K	ρc_p	$4{\cdot}10^5$ J/m^3
k_r	29732 s^{-1}	λ	0.8 W/m K
L	1 m	ϵ	0.8
Δt	100 s	T	1000 s

Appendix B

Algorithms

In this appendix we give the pseudo-code for some of the algorithms used in this thesis: the BSI rank $p+1$ continuation method as used in Chapter 5 and the optimal control method used in Chapter 6. The pseudo-codes for these algorithms go into more detail than the descriptions given in the respective chapters. They are meant to be a guideline for the interested reader who wants to implement some of the methods him/herself.

B.1 The BSI Rank $p+1$ Method

In this section we give the pseudo-codes for method discussed in Chapter 5. This method consists of a pseudo-arclength method in combination with the BSI rank $p+1$ method. First we give the pseudo-code for the pseudo-arclength continuation method with variable stepsize. The $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ needed as input in the pseudo-arclength code should both be solutions on the branch that we wish to continue. The δ in the input supplies an initial stepsize and δ_{max} supplies a maximum stepsize. In step 5 of the pseudo-arclength method the BSI rank $p+1$ method is called. The counter $it.BSI$ counts the number of period map evaluations, see the pseudo-core for the BSI rank $p+1$ method. If this number is to large, then the stepsize in the continuation algorithm is decreased.

Algorithm *Pseudo-arclength*
Input: $\delta \in \mathbb{R}$, $\delta_{max} \in \mathbb{R}$, $H_0 = I \in \mathbb{R}^{(N+1)\times(N+1)}$, $V_0 \in \mathbb{R}^{N\times p}$, $\boldsymbol{x}_0 = (x_0, \lambda_0) \in \mathbb{R}^{N+1}$ and $\boldsymbol{x}_1 = (x_1, \lambda_1) \in \mathbb{R}^{N+1}$

1. $s = 1$
2. **for** $i = 1, 2, 3, ...$
3. **do** $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + s(\boldsymbol{x}_i - \boldsymbol{x}_{i-1})$
4. $w = (x_i - x_{i-1})/|\boldsymbol{x}_i - \boldsymbol{x}_{i-1}|$
5. $\delta = s\delta$
6. call *BSI rank* $p+1(\boldsymbol{x}_{i+1}, V_i, H_i, w, it.BSI)$
7. **if** *BSI rank* $p+1$ failed
8. **then** $s = 0.5$

9. $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i$
10. $\boldsymbol{x}_i = \boldsymbol{x}_{i-1}$
11. **else if** $it.BSI < 100$ and $\delta < \delta_{max}$
12. **then** $s = \min(1.6, \delta_{max}/\delta)$
13. **else** print out $\boldsymbol{x}_{i+1}$ and the eigenvalues
14. s=1

The above given pseudo-code calls the BSI rank $p+1$ method. This methods computes the next periodic solution on the branch. The pseudo-code for the BSI rank $p+1$ method is given by

Algorithm *BSI rank* $p+1$
Input: $H_0 \in \mathbb{R}^{(N+1)\times(N+1)}$, $V_0 \in \mathbb{R}^{N\times p}$, $\boldsymbol{x}_0 = (x_0, \lambda_0) \in \mathbb{R}^{(N+1)}$ and $w \in \mathbb{R}^N$

1. $it.BSI = 0$
2. **for** $i = 1, 2, 3, ...$
3. **do** $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + H_i G(\boldsymbol{x}_i)$ ($*$ costs one evaluation of F $*$)
4. $it.BSI = it.BSI + 1$
5. Subspace iteration: compute $S := \frac{\partial F}{\partial x}(\boldsymbol{x}_{i+1})V_i$ ($*$ costs p evaluations of F $*$)
6. Compute the ordered Schur-factorization $Y^T R Y$ of $V_i S$. Compute $V = SY$. Orthonormalize the columns of V and put the result in V_{i+1}. $it.BSI = it.BSI + p$.
7. $$\boldsymbol{V} := \begin{pmatrix} V_i \\ 0 \end{pmatrix} \text{ and } \boldsymbol{Z} := \begin{pmatrix} \left(\frac{\partial F}{\partial x}(\boldsymbol{x}_{i+1}) - I\right)V_i \\ w^T V_i \end{pmatrix}.$$
8. $B_0 = H_i$
9. **for** $l = 0, ..., p-1$
10. **do** compute
$$B_{l+1} = \begin{cases} B_l - \dfrac{(\boldsymbol{v}_l + B_l \boldsymbol{z}_l)\boldsymbol{v}_l^T B_l}{\boldsymbol{v}_l^T B_l \boldsymbol{z}_l} & \text{if } \boldsymbol{v}_l^T B_l \boldsymbol{z}_l \neq 0, \\ B_l & \text{if } \boldsymbol{v}_l^T B_l \boldsymbol{z}_l = 0, \end{cases}$$
with $\boldsymbol{v}_l$ and $\boldsymbol{z}_l$ the lth column of respectively $\boldsymbol{V}$ and $\boldsymbol{Z}$
11. $\boldsymbol{p} = (I - \boldsymbol{V}\boldsymbol{V}^T)\,(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)$,
12. $\boldsymbol{f} = G(\boldsymbol{x}_{i+1}) - G(\boldsymbol{x}_i) - \boldsymbol{Z}\boldsymbol{V}^T(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)$.
13. Compute
$$H_{i+1} = \begin{cases} B_p - \dfrac{(\boldsymbol{p} + B_p \boldsymbol{f})\boldsymbol{p}^T B_p}{\boldsymbol{p}^T B_p \boldsymbol{f}} & \text{if } \boldsymbol{p}^T B_p \boldsymbol{f} \neq 0 \\ & \text{and } (I - \boldsymbol{Z}\boldsymbol{Z}^T)\big(G(\boldsymbol{x}_{i+1}) - G(\boldsymbol{x}_i)\big) \neq 0, \\ B_p & \text{otherwise,} \end{cases}$$
14. **until** $\boldsymbol{x}_i$ has converged or $it.BSI > 170$.
15. **return** $\boldsymbol{x}_i$, V_i, H_i, $it.BSI$ and if $it.BSI > 170$, return failure

B.2 The Optimal Control Algorithm

Here we present the pseudo-code for the optimal control algorithm used in Chapter 6. We first give the pseudo-code for the first order gradient method. This code uses the Newton-Picard method for the state equations and for the adjoint equations. These are two different routines for which we also given the pseudo-code. The code for the gradient method is:

Algorithm *Gradient Method*
Input: x_1, u_1, V_{p_1}, ϵ_1 and ϵ_2
1. **for** $i = 1, 2, 3, ...$
2. **do** call *Newton-Picard-state eq.*(x_i, V_{p_i}, p, S)
($*$ computes periodic solution of state equations $*$)
3. call *Newton-Picard-adjoint eq.*$(\Lambda, x_i, V_{p_i}, p, S)$
4. call *Newton-Picard-adjoint eq.*(R, x_i, V_{p_i}, p, S)
($*$ computes the periodic solutions $\Lambda(t)$ and $R(t)$ of the adjoint equations. $*$)
5. Compute the integrals

$$I_1 = \int_\Omega \left[R^T \frac{\partial f}{\partial u} + \frac{\partial M}{\partial u}\right]\left[R^T \frac{\partial f}{\partial u} + \frac{\partial M}{\partial u}\right]^T dt,$$
$$I_2 = \int_\Omega \left[\Lambda^T \frac{\partial f}{\partial u} + \frac{\partial L}{\partial u}\right]\left[R^T \frac{\partial f}{\partial u} + \frac{\partial M}{\partial u}\right]^T dt,$$
$$I_3 = \int_\Omega \left[\Lambda^T \frac{\partial f}{\partial u} + \frac{\partial L}{\partial u}\right]\left[\Lambda^T \frac{\partial f}{\partial u} + \frac{\partial L}{\partial u}\right]^T dt,$$

where $\Omega \subseteq [0,1]$ with $\Omega = \{t \in [0,1] \mid a < u_0(t) < b\}$.
6. compute

$$\Delta u(t) = -\epsilon_2 \Big[\frac{\partial L}{\partial u}(t, x(t), v(t)) + \xi \frac{\partial M}{\partial u}(t, x(t), v(t)) + [\Lambda(t) + R(t)\xi]^T \frac{\partial f}{\partial u}(t, x(t), v(t))\Big],$$

with

$$\xi = (\epsilon_2 I_1)^{-1}(\epsilon_1 \int_0^1 M(t, x(t), v(t))\, dt - \epsilon_2 I_2).$$

7. $u(t) = v(t) + \Delta u(t)$
8. $v(t) = u(t) + \max(a - u(t), 0) - \max(u(t) - b, 0)$.
($*$ This is done to make sure that the new control is in $[a, b]$. $*$)

In step 2 of the gradient method the Newton-Picard method for the state equations is used. This is the same method as presented in Chapter 3. The pseudo-code for this method is given by:

Algorithm *Newton-Picard-state eq.*
Input: x_0, V_0, p
1. **for** $i = 0, 1, 2, ...$
2. **do for** $j = 1, s$
3. **do** $S = J(x_i)V_{p_i}$
4. compute the ordered real Schur-factorization $Y^T RY$ of $V_{p_i}^T S$.
5. **if** $j < s$
6. **then** orthonormalize $V = SY$ and put the result in V_{p_i}.
7. **if** $|F(x_i) - x_i| < \delta_0$ **then** stop ($*$ 1 evaluation of F $*$).
8. $v_1 = (I - V_{p_i}V_{p_i}^T)(F(x_i) - x_i)$.
9. **for** $j = 1, 2..., l$ ($*$ Computation of $J(x_i)v_j$: l evaluations of F $*$).
10. **do** $v_{j+1} = (I - V_{p_i}V_{p_i}^T)(J(x_i)v_j + (F(x_i) - x_i)$.
11. $V_r\Delta\bar{r} := v_l$.
12. $S := J(x_i)V_{p_i}$ ($*$ p evaluations of F $*$).
13. Solve $(V_{p_i}^T S - I)(\Delta\bar{p}) + V_{p_i}^T J(x_i)(V_r\Delta\bar{r}) = -V_{p_i}^T(F(x_i) - x_i)$ for $\Delta\bar{p}$. ($*$ Computation of $J(x_i)(V_r\Delta\bar{r})$: 1 evaluation of F $*$).
14. $x_{i+1} := x_i + V_{p_i}\Delta\bar{p} + V_r\Delta\bar{r}$.
15. $V := SY$
16. orthonormalize V and put the result in $V_{p_{i+1}}$.
17. **return** x_i, V_{p_i} and S.

The Newton-Picard method for the adjoint equation is used in step 3 and in step 4 of the gradient method. In step 3 it is used to compute the periodic solution Λ and in step 4 to compute the periodic solution R. Here we give the pseudo-code for both steps only once. The only difference between the two cases is that the period map, denoted by K in the code below, should in the step 3 be the period map associated with the adjoint equation for Λ and in step 4 be the period map associated with the adjoint equation for R. Furthermore, in the pseudo-code below the x denotes the periodic solution computed of the state equations, computed with Newton-Picard method described above. This method also supplies the S and V_p. The y_1 is an initial approximation for the periodic solution Λ or R of the adjoint equations. The $J(x)$ in step 6 denotes the Jacobian of the period map of the state equation in the periodic state x.

Algorithm *Newton-Picard-adjoint eq.*
Input: V_p, y_1, x, S
1. **for** $i = 1, 2, 3, ...$
2. **if** $|K^{-1}(y_i) - y_i| < \delta_0$ **then** stop ($*$ 1 evaluation of K^{-1} $*$)

3. $\tilde{y}^T = (K^{-1}(y_i) - y_i)V_p(S - I)^{-1}$.
4. $\lambda^{(0)} = V_p\tilde{y}$
5. **for** $j = 0, 2, .., l - 1$
6. **do** $\lambda^{(j+1)} = J(x)^T\lambda^{(j)} - (K^{-1}(y_i) - y_i)$ $(* \ l$ evaluations of $K^{-1} \ *)$.
7. $y_{i+1} = y_i - \lambda^{(l)}$.
8. **return** y_i

Bibliography

[1] S.K. Bhatia. Analysis of catalytic reactor operation with periodic flow reversal. *Chem. Eng. Sci.*, 46:361–367, 1990.

[2] G. K. Boreskov and Y. S. Matros. Unsteady-state performance of heterogeneous catalytic reactions. *Catal. Rev. Sci. Engng.*, 25:552–569, 1983.

[3] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, 19:577–593, 1965.

[4] C. G. Broyden, J. E. Dennis, and J. J. Moré. On the local and superlinear convergence of quasi-Newton methods. *J. Inst. Math. Appl.*, 12:223–245, 1973.

[5] A. Bryson Jr. and Y. Ho. *Applied optimal control.* Washington, D.C., Hemisphere Publishing Company, 1975.

[6] Y. S. Cheng, E. Alpay, and L. S. Kerschenbaum. Simulation and optimisation of a rapid pressure swing reactor. *Comput. Chem. Engng.*, 22:S45–S52, 1998.

[7] D. T. Croft and M. G. Levan. Periodic states of adsorption cycles–I. Direct determination and stability. *Chem. Eng. Sci.*, 49:1821–1829, 1994.

[8] D. T. Croft and M. G. Levan. Periodic states of adsorption cycles–II. Solution space and multiplicity. *Chem. Eng. Sci.*, 49:1831–1841, 1994.

[9] M. M. Davis and M. D. Levan. Experiments on optimization of thermal swing adsorption. *Ind. Eng. Chem. Res.*, 28:778–785, 1989.

[10] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, 1983.

[11] Y. Q. Ding and M. D. LeVan. Periodic states of adsorption cycles iii. convergence acceleration for direct determination. *Chem. Eng. Sci.*, 56:5217–5230, 2001.

[12] G. Eigenberger and U. Nieken. Catalytic combustion with periodic flow reversal. *Chem. Eng. Sci.*, 43:2109–2115, 1988.

[13] D. M. Gay. Some convergence properties of Broyden's method. *SIAM J. Numer. Anal.*, 16:623–630, 1979.

[14] K. Georg. On tracing an implicitly defined curve by quasi-newton steps and calculating bifurcation by local perturbations. *SIAM J. Sci. Statist. Comput.*, 2:35–50, 1981.

[15] V. K. Gupta and S. K. Bhatia. Solution of cyclic profiles in catalytic reactor operation with periodic flow reversal. *Comput. Chem. Engng.*, 15:229–237, 1991.

[16] J. Hale. *Ordinary differential equations*. Wiley-Interscience, 1969.

[17] F. J. M. Horn and R. C. Lin. Periodic processes: a variational approach. *Ind. Eng. Chem. Process Design Develop.*, 6:21–30, 1967.

[18] Z. Huang. Restart row updates ABS methods for solving systems of nonlinear equations. *Computing*, 50:229–239, 1993.

[19] H. Jarausch and W. Mackens. Numerical treatment of bifurcation branches by adaptive condensation. In T. Küpper, H. Mittelman, and H. Weber, editors, *Numerical Methods for Bifurcation Problems*, ISNM 70, pages 296–309. Birhäuser-Verlag, 1984.

[20] H. Jarausch and W. Mackens. Solving large nonlinear systems of equations by an adaptive condensation process. *Numer. Math.*, 50:633–653, 1987.

[21] J. Khinast, A. Gurumoorthy, and D. Luss. Complex dynamic features of a cooled reverse-flow reactor. *AIChE Journal*, 44:1128–1140, 1998.

[22] J. Khinast and D. Luss. Mapping regions with different bifurcation diagrams of a reverse-flow reactor. *AIChE Journal*, 43:2034–2047, 1997.

[23] E. S. Kikkinides, V. I. Sikavitsas, and R. T. Yang. Natural gas desulfurization by adsorption: Feasibility and multiplicity of cyclic steady states. *Ind. Eng. Chem. Res.*, 34:255–262, 1995.

[24] E. S. Kikkinides and R. T. Yang. Simultaneous SO_2/NO_x removal and SO_2 recovery from flue gas by pressure swing adsorption. *Ind. Eng. Chem. Res.*, 30:1981–1989, 1991.

[25] N. F. Kirkby and J. E. P. Morgan. A theoretical investigation of pressure swing reaction. *Chem. Eng. Res. Des.*, 72:541–550, 1994.

[26] A. J. Kodde and A. Bliek. Selectivity enhancement in consecutive reactions using the pressure swing reactor. *Stud. Surf. Sci. Catal.*, 109:419–428, 1997.

[27] D. E. Kowler and R. H. Kadlec. The optimal control of a periodic adsorber. *AIChE Journal*, 18:1207–1219, 1972.

[28] Y. A. Kuznetsov. *Elements of applied bifurcation theory*, volume 112 of *Appl. Math. Sciences*. Springer-Verlag New York, Inc., 1995.

[29] H. M. Kvamsdal. *Studies on Modeling, Simulation and Optimization of PSA Systems.* PhD thesis, University of Trondheim, 1995.

[30] H. M. Kvamsdal and T. Hertzberg. Optimization of pressure swing adsorption systems – the effect of mass transfer during the blowdown step. *Chem. Eng. Sci.*, 50:1203–1212, 1995.

[31] H. M. Kvamsdal and T. Hertzberg. Optimization of PSA systems - studies on cyclic steady state convergence. *Comput. Chem. Engng.*, 21:819–832, 1997.

[32] H.M. Kvamsdal and T. Hertzberg. Pressure swing adsorption – optimization of a trace separation system. *Comput. Chem. Engng.*, 19:S339–S344, 1995.

[33] J. Lewandowski, N. O. Lemcoff, and S. Palosaari. Use of neural networks in the simulation and optimization of pressure swing adsorption processes. *Chem. Eng. Technol.*, 7:593–597, 1998.

[34] Y. Liu, J. Delgado, and J. A. Ritter. Comparison of finite difference techniques for simulating pressure swing adsorption. *Adsorption*, 4:337–344, 1998.

[35] K. Lust. *Numerical bifurcation analysis of periodic solutions of partial differential equations.* PhD thesis, Katholieke Universiteit Leuven, 1997.

[36] K. Lust and D. Roose. Computation and bifurcation analysis of periodic solutions of large-scale systems. In E. Doedel and L. S. Tuckerman, editors, *Numerical methods for bifurcation problems and large-scale dynamical systems (Minneapolis, MN, 1997)*, volume 119 of *IMA Vol. Math. Appl.*, pages 265–301, 2000.

[37] K. Lust, D. Roose, A. Spence, and A. R. Champneys. An adaptive Newton-Picard algorithm with subspace iteration for computing periodic solutions. *SIAM J. Sci. Comput.*, 19:1188–1209, 1998.

[38] S. Nilchan and C. C. Pantelides. On the optimisation of periodic adsorption processes. *Adsorption*, 4:113–147, 1998.

[39] J. Nocedal. Updating Quasi-Newton matrices with limited storage. *Math. Comp.*, 35:773–782, 1980.

[40] N. S. Raghavan, M. M. Hassan, and D. M. Ruthven. Numerical simulation of a PSA system. *AIChE Journal*, 31:385–395, 1985.

[41] S. U. Rege, J. Padin, and R. T. Yang. Olefin/paraffin separations by adsorption: π-complexation vs. kinetic separation. *AIChE Journal*, 44:799–809, 1998.

[42] R. C. Reid, J. M. Prausnitz, and B. E. Poling. *The properties of gases and liquids.* McGraw-Hill, Inc., 1987.

[43] D. Roose, K. Lust, A. Champneys, and A. Spence. A Newton-Picard shooting method for computing periodic solutions of large-scale dynamical systems. *Chaos, Solitons and Fractals*, 5:1913–1925, 1995.

[44] D. M. Ruthven and C. B. Ching. Countercurrent and simulated countercurrent adsorption separation processes. *Chem. Eng. Sci.*, 44:1011–1038, 1989.

[45] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Algorithms and Architectures for Advanced Scientific Computing, Manchester University Press, Manchester, 1992.

[46] A. G. Salinger and G. Eigenberger. The direct calculation of periodic states of the reverse flow reactor–I. Methodology and propane combustion results. *Chem. Eng. Sci.*, 51:4903–4913, 1996.

[47] A. G. Salinger and G. Eigenberger. The direct calculation of periodic states of the reverse flow reactor–II. Multiplicity and instability. *Chem. Eng. Sci.*, 51:4915–4922, 1996.

[48] J. M. Schork and J. R. Fair. Parametric analysis of thermal regeneration of adsorption beds. *Ind. Eng. Chem. Res.*, 27:457, 1988.

[49] G. Shroff and H. Keller. Stabilization of unstable procedures: The recursive projection method. *SIAM J. Numer. Anal.*, 30:1099–1120, 1993.

[50] O. J. Smith IV and A. W. Westerberg. Acceleration of cyclic steady state convergence for pressure swing adsorption models. *Ind. Eng. Chem. Res.*, 31:1569–1573, 1992.

[51] E. Spedicato and Z. Huang. Numerical experience with Newton-like methods for nonlinear algebraic systems. *Computing*, 58:69–89, 1997.

[52] F. Stepanek, M. Kubicek, and M. Marek. Modeling and optimization of continuous adsorption in a dessicant rotor. *Ind. Eng. Chem. Res.*, 37:1435–1443, 1998.

[53] L. M. Sun and P. Le Queré. Numerical simulation of diffusion-limited PSA process models by finite difference methods. *Chem. Eng. Sci.*, 51:5341–5352, 1996.

[54] N. Sundaram. A noniterative solution of periodic states in gas purification pressure swing adsorption. *Ind. Eng. Chem. Res.*, 32:1686–1691, 1993.

[55] N. Sundaram and R. T. Yang. On the pseudomultiplicity of pressure swing adsorption periodic states. *Ind. Eng. Chem. Res.*, 37:154–158, 1998.

[56] J. Unger, G. Kolios, and G. Eigenberger. On the efficient simulation and analysis of regenerative processes in cyclic operation. *Comput. Chem. Engng.*, 21:S167–S172, 1997.

[57] T. L. Van Noorden, S. M. Verduyn Lunel, and A. Bliek. Direct determination of cyclic steady states in cyclically operated packed bed reactors. In F. Keil, W. Mackens, H. Voss, and J. Werter, editors, *Scientific Computing in Chemical Engineering II*, volume 1, pages 311–318. Springer, 1999.

[58] T. L. Van Noorden, S. M. Verduyn Lunel, and A. Bliek. A Broyden rank $p + 1$ update continuation method with subspace iteration. Technical Report MI 2001-30, Mathematical Institute, University of Leiden, 2001.

[59] T. L. Van Noorden, S. M. Verduyn Lunel, and A. Bliek. The efficient computation of periodic states of cyclically operated chemical processes. Technical Report MI 2001-22, Mathematical Institute, University of Leiden, 2001.

[60] T. L. Van Noorden, S. M. Verduyn Lunel, and A. Bliek. Optimization of cyclically operated reactors and separators. Technical Report MI 2001-31, Mathematical Institute, University of Leiden, 2001.

[61] T. L. Van Noorden, S. M. Verduyn Lunel, and A. Bliek. Acceleration of the determination of periodic states of cyclically operated reactors and separators. *Chem. Eng. Sci.*, 57:1041–1055, 2002.

[62] T.L. Van Noorden, S. M. Verduyn Lunel, and A. Bliek. Direct determination of periodic states of cyclically operated packed bed reactors. In G. F. Froment and K. C. Waugh, editors, *Reaction Kinetics and the Development and Operation of Catalytic Processes*, volume 133 of *Studies in Surface Science and Catalysis*, pages 263–270. Elsevier, 2001.

[63] G. G. Vaporciyan and R. H. Kadlec. Periodic separating reactors: experiments and theory. *AIChE Journal*, 35:831–844, 1989.

[64] R. T. Yang. *Gas separation by adsorption processes.* Butterworths, Stoneham, MA, 1987.

Nederlandse samenvatting

Nieuwe algoritmen voor parameter-swing reactoren

Na het opstarten van een chemisch proces wordt in de regel eerst een fase van voorbijgaande aard doorlopen. Vervolgens zal het proces in de buurt komen van een limiet-toestand. Als het proces deze limiet-toestand voldoende dicht heeft genaderd, dan zal het, zolang het in werking is, in de omgeving van deze limiet-toestand blijven, en de limiet-toestand steeds dichter naderen. De werking van het proces in de omgeving van deze limiet-toestand is de basis voor het ontwerpen en het optimaliseren van chemische processen. Voor het bestuderen en analyseren van chemische processen is het daarom van belang dat de limiet-toestanden van processen zo efficiënt mogelijk bepaald kunnen worden.

Traditioneel worden chemische processen zo ontworpen dat de operatie-condities constant worden gehouden gedurende de werking van het proces. Dit soort processen worden *steady state operated* processen genoemd. Naast deze *steady state operated* processen zijn er nu meer en meer processen waarbij de operatie-condities cyclisch in de tijd variëren. Dit soort processen heten *cyclische* processen.

Voor *steady state operated* processen zijn de limiet-toestanden vaak tijdsinvariant. We noemen zo'n toestand een *steady state*. In termen van modelvergelijkingen, kan een *steady state* vaak beschreven worden door een oplossing van een stelsel algebraïsche vergelijkingen of door oplossing van een stelsel tijdsonafhankelijke gewone differentiaalvergelijkingen. De afgelopen decennia is veel theorie ontwikkeld – veelal numeriek – om de oplossing van zulke differentiaalvergelijkingen te bestuderen. Dit maakt het mogelijk de limiet-toestanden van *steady state operated* processen doeltreffend te analyseren.

Door de periodieke aandrijving door middel van de cyclisch variërende operatie-condities, kan een cyclisch proces nooit een tijdsinvariante toestand bereiken. Een limiet-toestand van een cyclisch proces in daarom intrinsiek dynamisch van aard. De typische limiet-toestand van een cyclisch proces in een zo genaamde *cyclus-invariante* of *periodieke* toestand. In termen van modelvergelijkingen, kan een periodieke toestand beschreven worden door een oplossing van een twee-puntsrandwaarde probleem voor een groot stelsel van differentiaalvergelijkingen. Door de tijdsafhankelijke karakteristieken, zijn de theoretische bestudering en numerieke benadering van een periodieke toestand van een cyclisch proces veel minder eenvoudig dan de bepaling van een *steady state* van een *steady state operated* chemisch proces.

In dit proefschrift bestuderen we cyclische chemische processen en het efficiënt numeriek bepalen van hun periodieke toestanden. Modellen van cyclische processen worden vaak gegeven in termen van stelsels van partiële differentiaalvergelijkingen met coëfficiënten en randvoorwaarden die periodiek veranderen in de tijd. Om deze modellen numeriek te kunnen bestuderen, moeten we eerst de modelvergelijkingen discretiseren. Dit gaat in de regel als volgt. Eerst worden de ruimte-variabelen gediscretiseerd. We verkrijgen dan een groot stelsel van N (waar N van de discretisatie af hangt) gewone differentiaalvergelijkingen. Dit stelsel kunnen we abstract schrijven als

$$x'(t) = f(t, x(t)), \quad \text{waar} \quad f(t+T, \cdot) = f(t, \cdot), \tag{B.1}$$

waar $x(t)$ een N-dimensionale vector is en, $f(t, \cdot)$ een functie van de ruimte van N-dimensionale vectoren naar zichzelf. Een periodieke oplossing van (B.1) is een oplossing met de eigenschap dat $x(0) = x(T)$.

De afbeelding F die aan de beginwaarde $x(0)$ de waarde van de oplossing op het tijdstip $t = T$, $x(T)$ toekent, noemen we de *Poincaré* afbeelding of *periode*-afbeelding. Deze afbeelding kunnen we dus schrijven als

$$F(x_0) = x(T, x_0),$$

waar $x(t, x_0)$ de oplossing van vergelijking (B.1) is, met beginwaarde $x(0) = x_0$. We zien dat de beginwaarde van een periodieke oplossing van (B.1) een dekpunt is van de de periode-afbeelding: $F(x_0) = x(T, x_0) = x(0, x_0) = x_0$.

Een dekpunt x^* van F wordt *stabiel* genoemd als het de beginwaarde is van een aantrekkende periodieke oplossing van (B.1). Met een aantrekkende periodieke oplossing bedoelen we een periodieke oplossing waarvoor dichtbij gelegen oplossingen dichtbij blijven en ook naar de periodieke oplossing convergeren. De stabiliteit van een dekpunt x^* wordt bepaald door eigenschappen van de Jacobiaan $J(x^*)$, of afgeleide, van F. Indien alle eigenwaarden van $J(x^*)$ absolute waarde kleiner dan één hebben, dan is het dekpunt x^* stabiel, en als er een eigenwaarde is met absolute waarde groter dan één dan is x^* instabiel.

De eenvoudigste manier om een stabiel dekpunt van een afbeelding te bepalen, is de Picard iteratie van de afbeelding. Deze methode komt in het geval van een cyclisch proces beschreven door (B.1) overeen met het dynamisch simuleren van het proces met een gegeven initiële toestand. De Picard iteratie zal dus, net als de begintoestanden van de cycli van het chemische proces, eerst de initiële fase van voorbijgaande aard moeten doorlopen, voordat de iteratie de periodieke limiet-toestand voldoende goed benaderd heeft. Voor sommige cyclische processen kan deze initiële fase uit duizenden cycli, en dus ook uit duizenden Picard iteraties, bestaan voordat een periodieke toestand voldoende goed benaderd is [23]. Elke Picard iteratie komt overeen met één functie-evaluatie van F. Aangezien een functie-evaluatie van F veel rekenwerk kost, is de Picard iteratie van F in veel gevallen een tijdrovende methode om dekpunten van F te bereken.

Om het aantal functie-evaluaties van F te proberen te beperken, kunnen we ook de Newton methode gebruiken om dekpunten van F te bepalen. De Newton methode [7] heeft

in het algemeen minder iteraties nodig om naar een dekpunt van de periode-afbeelding F te convergeren dan de Picard iteratie van F. Bovendien kan de Newton methode ook instabiele dekpunten van F berekenen. Voor elke iteratie van de Newton methode is het wel nodig om de afgeleide van F te berekenen. Deze $N \times N$ matrix kan met behulp van eindige differenties berekend worden, maar hiervoor zijn ten minste $N + 1$ functie-evaluaties van F nodig. Elke iteratie van de Newton methode vergt dus ongeveer N maal zoveel rekenwerk als een Picard iteratie.

De Broyden [50] methode is een poging om het opnieuw berekenen van de Jacobiaan in iedere iteratie van de Newton methode te vermijden. De Broyden methode start met een benadering van de Jacobiaan, die in elke iteratie aangepast wordt met behulp van de informatie die is verkregen door F in de nieuwe benadering van de periodieke oplossing te evalueren. De Broyden methode kan, net als de Newton methode, zowel stabiele als instabiele dekpunten van F berekenen, maar de Broyden methode kan niet zelf de stabiliteit van de berekende periodieke toestand bepalen. Eén iteratie van de Broyden methode vergt slechts één functie-evaluatie van F, ongeveer evenveel rekenwerk als een Picard iteratie, en dus veel minder dan een iteratie van de Newton methode.

Alhoewel het efficiënt bepalen van periodieke toestanden enorm in de belangstelling staat, zijn een aantal fundamentele vragen nog onbeantwoord. In de chemische literatuur is vooral de Broyden methode met veel succes toegepast om snel periodieke toestanden te berekenen. We kunnen ons afvragen of er een verklaring is waarom de Broyden methode zo succesvol is. En is het mogelijk om voor een gegeven model van een cyclisch proces van te voren te zeggen welke methode het efficiëntst zal zijn? Om deze vraag doeltreffend te beantwoorden moet ook rekening gehouden worden met het doel van de berekeningen. Wil men periodieke oplossingen bepalen om het proces te optimaliseren of om een bifurcatie-analyse van het proces uit te voeren? Een andere gerelateerde vraag is of we eigenschappen van het onderliggende chemische proces kunnen gebruiken om snellere methoden te ontwikkelen. In dit proefschrift komen deze vragen uitvoerig aan de orde. We hebben nog niet een volledig antwoord op alle vragen, maar leveren wel een bijdrage aan het uiteindelijke antwoord.

In Hoofdstuk 3 presenteren we numerieke berekeningen die aangeven dat de bestudeerde cyclische processen laag dimensionale dynamica vertonen. Hiermee bedoelen we dat het aantal grote eigenwaarden van de Jacobiaan $J(x)$ van de periode-afbeelding F berekend in verschillende punten x klein is, en, in principe, onafhankelijk van de gebruikte discretisatie. Dit betekent dat de interessante en stabiliteit bepalende dynamica plaats vindt in een laag dimensionale deelruimte van de hele toestandsruimte.

We introduceren een *single shooting* methode in combinatie met een hybride Newon-Picard methode, ontwikkeld door Lust et al. in [37]. Deze methode buit de laag dimensionale dynamica van de chemische processen uit voor het efficiënt berekenen van hun periodieke toestanden. De Newton-Picard methode gebruikt het feit dat er slechts een klein aantal grote eigenwaarden zijn om de expliciete berekening van de Jacobiaan $J(x)$ van F te vermijden. We hoeven slechts de actie van $J(x)$ op een p-dimensionale deelruimte te bepalen, waarbij p veel kleiner is dan N. We bespreken de Newton-Picard methode en

passen de methode toe om periodieke toestanden van twee cyclische processen te berekenen: van een *rapid pressure swing adsorber* en een *reverse flow reactor*. Als een illustratie van de efficiëntie van de methode berekenen we een tak periodieke toestanden van de *reverse flow reactor*.

In Hoofdstuk 4 vergelijken we de prestatie van de Newton-Picard methode met die van de Picard iteratie methode, de Newton methode en van de Broyden methode. We passen deze vier methoden toe om periodieke toestanden van vijf goed bestudeerde en goed gedocumenteerde cyclische processen te berekenen. Om de methoden te kunnen vergelijken, introduceren we twee verschillende convergentie-criteria, één gebaseerd op het verschil tussen de begin- en eindtoestand van een cyclus (het residu) en één gebaseerd op de afstand van een toestand tot de periodieke toestand. Dit laatste convergentie-criterium heeft als nadeel dat de periodieke toestand al bekend moet zijn om het criterium te kunnen toepassen. Twee van de vijf processen (de H_2O/lucht en CO_2/N_2 pressure swing adsorption processen) vertonen lineair convergentie-gedrag. Voor deze processen kunnen we *a priori* voorspellingen maken over de prestaties van de verschillende methoden.

We vinden dat de Newton-Picard methode in de meeste gevallen efficiënter is dan zowel Picard iteratie als de Newton methode. De Broyden methode is in de geteste gevallen altijd het efficiëntst in termen van het aantal gebruikte evaluaties van de periode-afbeelding.

Bij de vergelijking tussen de Broyden methode en de Newton-Picard methode, moet men zich realiseren dat de Newton-Picard methode meer informatie over het bestudeerde cyclische proces geeft dan de Broyden methode. De Newton-Picard methode bepaalt immers ook de eigenwaarden van de Jacobiaan in de berekende periodieke toestand. Dit betekent dat de Newton-Picard methode een betere keus is dan de Broyden methode in gevallen waar het van belang is om ook de stabiliteit van periodieke oplossingen te bepalen.

Hoofdstuk 5 is gewijd aan het efficiënt berekenen van takken van periodieke toestanden van cyclische processen. We gebruiken een *pseudo-arclength* continueringsprocedure en we willen ook de stabiliteit bepalen van de berekende periodieke toestanden langs de tak. De Newton-Picard methode is zeer geschikt voor deze taak. Broyden's methode is, zoals we hierboven al opmerkten, niet geschikt om de stabiliteit van periodieke toestanden te bepalen. In dit hoofdstuk passen we echter de Broyden methode aan om toch de nodige eigenwaarden te kunnen benaderen. We gebruiken deze nieuwe methode, die we de Broyden rank $p+1$ update continuation method with Subspace Iteration noemen, of in het kort BSI rank $p+1$ methode, in combinatie met een pseudo-arclength procedure en vergelijken de prestatie van de BSI rank $p+1$ methode met de prestatie van de Newton-Picard methode. Met beide methoden berekenen we een tak van periodieke toestanden van een *reverse flow reactor*. We laten zien dat de BSI rank $p+1$ methode de efficiëntste methode van de twee is: deze methode heeft ongeveer de helft minder functie-evaluaties van de periode-afbeelding F nodig dan de Newton-Picard methode.

In Hoofdstuk 6 bekijken we de optimalisatie van een *rapid pressure swing adsorber* en een *rapid pressure swing reactor*. We optimaliseren deze twee processen met betrekking tot tijdsafhankelijke parameters (*optimal control*). We gebruiken een eerste orde gradiënt

methode om een oplossing van het optimal control probleem te benaderen. We laten zien dat de Newton-Picard methode in combinatie met de gradiënt methode zeer efficiënt werkt.

Voor de *rapid pressure swing adsorber* minimaliseren we de energie die het proces nodig heeft en voor de *rapid pressure swing reactor* maximaliseren we de opbrengst. Voor beide processen gebeurt de optimalisatie onder de voorwaarde dat het product-gas een minimale zuiverheid behoudt.

Voor beide besproken processen bestaat de operatie-procedure gewoonlijk uit twee stappen: de *pressurization* stap en de *depressurization* stap. De berekende oplossingen van de optimal control problemen laten echter zien dat voor zowel de *rapid pressure swing adsorber* en de *rapid pressure swing reactor* de optimale operatie-procedure aanleiding geeft tot een design met vier, in plaats van de oorspronkelijke twee, stappen: een *pre-pressurization* step, een *pressurization* stap, een *post-pressurzation* stap en een *depressurization* stap. In deze optimale operatie-procedures moet de voedingsdruk continu in de tijd gevariëerd worden. We laten zien dat de optimale procedures goed benaderd kunnen worden met vier-staps-procedures met stuksgewijs constante voedingsdruk.

Dankwoord

Op deze plek wil ik iedereen bedanken die mij op de een of andere manier heeft bijgestaan op de weg naar de voltooiing van dit proefschrift. Allereerst zijn dat mijn beide begeleiders en promotoren: Sjoerd Verduyn Lunel en Alfred Bliek. Zonder onze vruchtbare samenwerking was dit proefschrift er niet geweest.

De leescommissie-leden W.H. Hundsdorfer, M. van Veldhuizen, R. Krishna, D. Roose en C.C. Pantelides wil ik bedanken voor het aandachtig en kritisch lezen van mijn proefschrift en voor de daaruit volgende waardevolle opmerkingen en vragen.

Gedurende mijn tijd als aio heb ik veel heen en weer gefietst tussen een aantal werkplekken. Dat waren de Analyse groep op de VU en het Instituut voor Technische Scheikunde op de UvA, en later, na de verhuizing van Sjoerd naar Leiden, werd het Mathematisch Instituut in Leiden een beetje aan mijn lijstje van werkplekken toegevoegd.

Ik wil alle medewerkers en collega's op deze werkplekken bedanken voor de goede tijden en prettige werksfeer. Met name wil ik mijn kamergenoten noemen. Op de VU waren dat de bewoners van kamer R3.57: met Bas Lemmens en Derk Pik heb ik op de VU erg veel plezier gehad, met Ronald Geskus, Damien White en Alistair Vardy was het delen van de kamer van kortere duur, maar daarom niet minder aangenaam.

Op de UvA stond mijn bureau op kamer C6.16'. Die kamer deelde ik met veel genoegen met Ralph Jacobs, Adriaan Kodde, Bart de Graaf en Jasper Heuft. Adriaan en Bart hebben ook veel voor mij betekend op mijn pad door de Wondere Wereld der Technische Scheikunde.

In Leiden kwam ik weer een aantal oude bekenden tegen. Ik bedank Bart van de Rotten en Onno van Gaans, ze zorgden ervoor dat ik mij in Leiden meteen thuis voelde.

De steun en liefde van familie en vrienden heeft mij enorm geholpen bij het voltooien van dit proefschrift. Ik wil de familie Kruisselbrink bedanken voor hun warmte en vriendschap. Mijn ouders, Topy en Leon, mijn broer Erik en mijn zus Raven wil ik bedanken voor de verbondenheid die ik met hen voel. Als laatste bedank ik Astrid, die als geen ander weet wat dit proefschrift voor mij betekent.